

[illegible]

```
LL      000000  CCCCCCCC  KK      KK
LL      000000  CCCCCCCC  KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KKKKKK  KK
LL      00      00  CC      KKKKKK  KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LL      00      00  CC      KK      KK
LLLLLLLL 000000  CCCCCCCC  KK      KK
LLLLLLLL 000000  CCCCCCCC  KK      KK
                     ....
                     ....
                     ....
                     ....
```

```
LL      111111  SSSSSSSS
LL      111111  SSSSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SSSSSS
LL      11      SSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LLLLLLLL 111111  SSSSSSSS
LLLLLLLL 111111  SSSSSSSS
```

(1)	2	COPYRIGHT NOTICE
(1)	29	PROGRAM DESCRIPTION
(2)	77	DECLARATIONS
(3)	103	STORAGE DEFINITIONS
(5)	342	READ-ONLY DATA DEFINITIONS
(6)	465	SHOW_ALL_LOCKS - ACTION ROUTINE TO DISPLAY ALL LOCKS
(7)	499	SHOW_ONE_LOCK - ACTION ROUTINE TO DISPLAY ONE LOCK
(8)	529	SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
(9)	572	GET_CKB - GET LOCK BLOCK
(10)	618	SAVE_LOCK_DATA - RETRIEVE LOCK DATA FROM SYSTEM
(11)	658	DISPLAY_LOCK - CONTROL DISPLAY OF LOCK DATA
(12)	683	FORMAT_LOCK - FORMAT LOCK DATA
(13)	867	PRINT_LOCK - OUTPUT LOCK DATA
(14)	990	LOCK_COND_HAND - CONDITION HANDLER FOR SHOW_ALL_LOCKS
(15)	1029	SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY ALL RESOURCES
(16)	1091	SHOW_ONE_RES - ACTION ROUTINE TO DISPLAY ONE RESOURCE
(17)	1124	SHOW_RSB - CONTROL FOR RSB DISPLAY
(18)	1147	SHOW_QUEUES - DISPLAY QUEUES FOR GIVEN RESOURCE
(19)	1180	FORMAT_RSB - FORMAT RSB DATA
(20)	1257	PRINT_RSB - OUTPUT RSB DATA
(21)	1358	PROCESS_QUEUE - TRAVERSE RESOURCE QUEUES
(22)	1428	PRINT_LINE - OUTPUT QUEUE DATA

```
0000 1      .TITLE  LOCK LOCK AND RESOURCE FORMATTING ROUTINES
0000 2      .SBTTL  COPYRIGHT NOTICE
0000 3      .IDENT  'V04-000'
0000 4      :
0000 5      :*****
0000 6      :
0000 7      :  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :  TRANSFERRED.
0000 17     :
0000 18     :  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :  CORPORATION.
0000 21     :
0000 22     :  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :*****
0000 26     :
0000 27     :
```



```
0000 29 .SBTTL PROGRAM DESCRIPTION
0000 30 :++
0000 31 FACILITY
0000 32
0000 33 SYSTEM DUMP ANALYZER
0000 34
0000 35 ABSTRACT
0000 36
0000 37 ROUTINES TO FORMAT LOCK AND RESOURCE INFORMATION
0000 38
0000 39 ENVIRONMENT
0000 40
0000 41 NATIVE MODE, USER MODE
0000 42
0000 43 AUTHOR
0000 44
0000 45 MARYANN HINDEN, JUNE 1982
0000 46
0000 47 MODIFIED BY
0000 48
0000 49 V03-007 SRB0109 Steve Beckhardt 2-Feb-1984
0000 50 Fixed formatting of Directory entry text in SHO RES line.
0000 51
0000 52 V03-006 SRB0107 Steve Beckhardt 7-Dec-1983
0000 53 Enhanced SHOW LOCK and SHOW RESOURCE to display
0000 54 distributed lock information.
0000 55
0000 56 V03-005 SRB0089 Steve Beckhardt 27-May-1983
0000 57 Fixed bug displaying resource name. Added new status bit
0000 58 definitions.
0000 59
0000 60 V03-004 SRB0071 Steve Beckhardt 16-Mar-1983
0000 61 Changed the way system resources are decoded in RSB
0000 62 to use zero group number instead of SYSNAM flag.
0000 63
0000 64 V03-003 MSH0003 Maryann Hinden 17-Dec-1982
0000 65 Fix broken assume's.
0000 66
0000 67 V03-002 MSH0002 Maryann Hinden 22-Oct-1982
0000 68 Determine empty lock queue correctly in SHOW_PROC_LOCK.
0000 69 Change lock display - output lock and parent id in hex;
0000 70 rearrange format.
0000 71
0000 72 V03-001 MSH0001 Maryann Hinden 16-Jul-1982
0000 73 Correct references to LCK$xxxx data structures; change
0000 74 condition handling.
0000 75 :--
```

```
0000 77 .SBTTL DECLARATIONS
0000 78 :
0000 79 : SYSTEM SYMBOL DEFINITIONS
0000 80 :
0000 81 :
0000 82 $CHFDEF
0000 83 $LCKDEF
0000 84 $LKBDEF
0000 85 $OPTDEF
0000 86 $PCBDEF
0000 87 $RSBDEF
0000 88 $STSDEF
0000 89 :
0000 90 : LOCAL SYMBOL DEFINITIONS
0000 91 :
0000 92 :
0000 93 :
FFFFFFFF 0000 94 GRANT = -1 ; SYMBOLIC DESIGNATIONS FOR QUEUE TYPES
00000000 0000 95 CONVERT = 0
00000001 0000 96 WAIT = 1
0000 97 :
00000000 0000 98 LKID_OFF = LKB$L_LKID-LKB$L_LKID ; OFFSETS INTO SAVED QUEUE DATA
00000005 0000 99 GRMD_OFF = LKB$B_GRMODE-LKB$L_LKID
00000004 0000 100 RQMD_OFF = LKB$B_RQMODE-LKB$L_LKID
0000 101
```

```
0000 103 .SBTTL STORAGE DEFINITIONS
00000000 104 .PSECT SDADATA,NOEXE,WRT,LONG
0000 105 :
0000 106 : Data area for SHOW LOCKS commands
0000 107 :
0000 108 :
0000 109 LOCK_COUNT: ; count of locks processed
00000000 0000 110 .LONG 0
0004 111 PARID_BFR: ; parent lock id
00000000 0004 112 .LONG 0
0008 113 LKB_BFR: ; lock block data
00000068 0008 114 .BLKB LKB$K_LENGTH
0068 115 :
0000006C 0068 116 RSBCSID_BFR: ; CSID in RSB
0068 117 .BLKL 1
006C 118 LKB_RSB_BFR: ; resource block data for given lock
006C 119 GROOP_BFR: ; group number
0000006E 006C 120 .BLKW 1
006E 121 RMOD_BFR: ; access mode
0000006F 006E 122 .BLKB 1
006F 123 RSNLEN_BFR: ; resource name length
00000070 006F 124 .BLKB 1
0070 125 RESNAM_BFR: ; resource name
00000090 0070 126 .BLKB RSB$K_MAXLEN+1
0090 127 :
00000094 0090 128 LOCKID:: ; lock ID currently being processed
0090 129 .BLKL 1
0094 130 :
0094 131 .ALIGN LONG
0094 132 :
0094 133 FAO_RMINFO_DSC:
0094 134 .ASCID "of lock !XL on system !XL"
00B5 135 :
00B5 136 .ALIGN LONG
00B8 137 RMINFO_LEN:
00B8 138 .LONG 0
00BC 139 RMINFO_DSC:
00BC 140 .LONG 80
00000050 00BC 141 .ADDRESS RMINFO_BFR
000000C5 00C0 142 :
00C4 143 RMINFO_CNT:
00C4 144 .BLKB 1
00C5 145 RMINFO_BFR:
00C5 146 .BLKB 80
0115 147 :
0115 148 : Data area for SHOW RESOURCES commands
0115 149 :
0115 150 :
0115 151 :
0115 152 RES_COUNT: ; count of resources processed
00000000 0115 153 .LONG 0
0119 154 QUEUE_COUNT: ; count of queue elements processed
00000000 0119 155 .LONG 0
011D 156 RSB_BFR: ; resource block data
0000018C 011D 157 .BLKB RSB$K_MAXLEN+RSB$K_LENGTH
```

```
63 6F 6C 20 66 6F 0000009C 010E0000
73 79 73 20 6E 6F 20 4C 58 21 20 6B
4C 58 21 20 6D 65 74 00AE
```



```
00000190 018C 158 HTBL_INDx: ; current index into hash table
018C 159 .BLKL 1
00000194 0190 160 HTBL_CNT: ; size of hash table
0190 161 .BLKL 1
00000198 0194 162 COUNT: ; # queue items to display
0194 163 .BLKL 1
0000019C 0198 164 QUEUE_TYPE: ; type of queue being processed
0198 165 .BLKL 1
0000019F 019C 166 GRMD_BFR: ; grant mode
019C 167 .BLKB 3
000001A2 019F 168 RQMD_BFR: ; requested mode
019F 169 .BLKB 3
000001AE 01A2 170 LKID_BFR: ; lock id
01A2 171 .BLKL 3
000001AE 01AE 172 FAO_GROUP_DSC:
01AE 173 .ASCID "'!30W'"
57 4F 33 21 000001B6'010E0000' 01BA 174
01BA 175 GROUP_BUF:
000001BD 01BA 176 .BLKB 3
01BD 177
01BD 178 .ALIGN LONG
00000003 01C0 179 GROUP_BUF_DSC:
01C0 180 .LONG 3
000001BA' 01C4 181 .ADDRESS GROUP_BUF
01C8 182
01C8 183 SYS_DSC:
6D 65 74 73 79 53 000001D0'010E0000' 01C8 184 .ASCID "System"
01D6 185
01D6 186 .ALIGN LONG
0000000B 01D8 187 GROUP_DSC:
01D8 188 .LONG 11
000001E0' 01DC 189 .ADDRESS GROUP_TXT
01E0 190
01E0 191 GROUP_TXT:
20 20 20 70 75 6F 72 47 01E0 192 .ASCII "Group "
01E8 193 GROUP_NUM:
000001EB 01E8 194 .BLKB 3
01EB 195
01EB 196 ;
01EB 197 ; FAO data storage for RESOURCE display
01EB 198 ;
01EB 199
01EB 200 ; Line 1
01EB 201
00000000 01EB 202 RSB: .LONG 0 ; Address of RSB
00000000 01EF 203 GGMOD: .LONG 0 ; Group grant mode (addr of ASCII string)
01F3 204
01F3 205 ; Line 2
01F3 206
00000000 01F3 207 PRSB: .LONG 0 ; Address of parent RSB
00000000 01F7 208 CGMOD: .LONG 0 ; Conversion grant mode (addr of ASCII string)
01FB 209
01FB 210 ; Line 3
01FB 211
00000000 01FB 212 SRSBCT: .LONG 0 ; Sub-RSB reference count
00000000 01FF 213 BAST: .LONG 0 ; Blocking AST count
0203 214
```



```
0203 215 ; line 4
0203 216
00000000 0203 217 VAL4: .LONG 0 ; Value block longword #4
00000000 0207 218 VAL3: .LONG 0 ; Value block longword #3
00000000 020B 219 VAL2: .LONG 0 ; Value block longword #2
00000000 020F 220 VAL1: .LONG 0 ; Value block longword #1
00000000 0213 221 SEQNUM: .LONG 0 ; Sequence number
00000000 0217 222 VALID: .LONG 0 ; Value block valid (addr. of .ASCII string)
021B 223
021B 224 ; line 5
021B 225
00000000 021B 226 RESN2: .LONG 0 ; Second longword of resource name
00000000 021F 227 RESN1: .LONG 0 ; First
00000008 0223 228 .LONG 8 ; Count of ASCII string
00000000 0227 229 BUF1: .LONG 0 ; Text of resource name (addr of ASCII string)
022B 230
022B 231 ; line 6
022B 232
00000000 022B 233 RESNLEN: .LONG 0 ; Resource name length
00000000 022F 234 RESN4: .LONG 0 ; Fourth longword of resource name
00000000 0233 235 RESN3: .LONG 0 ; Third
00000008 0237 236 .LONG 8 ; Count of ASCII string
00000000 023B 237 BUF2: .LONG 0 ; Text of resource name (addr of ASCII string)
00000000 023F 238 CSID: .LONG 0 ; RSB CSID
0243 239
0243 240 ; line 7
0243 241
00000000 0243 242 RACMOD: .LONG 0 ; Resource access mode (addr of ASCII string)
00000000 0247 243 RESN6: .LONG 0 ; Sixth longword of resource name
00000000 024B 244 RESN5: .LONG 0 ; Fifth
00000008 024F 245 .LONG 8 ; Count of ASCII string
00000000 0253 246 BUF3: .LONG 0 ; Text of resource name (addr of ASCII string)
0257 247 DIRENTRY:
00000000 0257 248 .LONG 0 ; Directory entry (addr. of .ASCII string)
025B 249
025B 250 ; line 8
025B 251
00000000 025B 252 RNSPACE: .LONG 0 ; Address of descriptor
00000000 025F 253 RESN8: .LONG 0 ; Eighth longword of resource name
00000000 0263 254 RESN7: .LONG 0 ; Seventh
00000008 0267 255 .LONG 8 ; Count of ASCII string
00000000 026B 256 BUF4: .LONG 0 ; Text of resource name (addr of ASCII string)
026F 257
```

```
026F 259 :  
026F 260 :      FAO data storage for LOCKS display  
026F 261 :  
026F 262 :  
026F 263 : LINE 1  
026F 264 :  
00000000 026F 265 LKID:  .LONG  0      : Lock id (value)  
00000000 0273 266 PID:  .LONG  0      : PID (value)  
00000000 0277 267 FLAGS1: .LONG  0      : First line of flags (addr. of .ASCII string)  
00000000 027B 268         .LONG  0      :  
00000000 027F 269         .LONG  0      :  
0283 270 :  
0283 271 : LINE 2  
0283 272 :  
00000000 0283 273 PARID:  .LONG  0      : Parent id (value)  
00000000 0287 274 STATE1: .LONG  0      : Lock state info (address of .ASCII string)  
00000002 028B 275         .LONG  2      : Lock mode (length of string)  
00000000 028F 276 MODE1:  .LONG  0      : Lock mode (address of .ASCII string)  
00000000 0293 277 FLAGS2: .LONG  0      : Second line of flags (addr. of .ASCII string)  
00000000 0297 278         .LONG  0      :  
00000000 029B 279         .LONG  0      :  
029F 280 :  
029F 281 : LINE 3  
029F 282 :  
00000000 029F 283 SUBLKS: .LONG  0      : Sublocks (value)  
00000000 02A3 284 STATE2: .LONG  0      : More lock state info (addr. of .ASCII string)  
00000002 02A7 285         .LONG  2      : More lock mode (length of string)  
00000000 02AB 286 MODE2:  .LONG  0      : More lock mode (addr. of .ASCII string)  
00000000 02AF 287 FLAGS3: .LONG  0      : Third line of flags (addr. of .ASCII string)  
00000000 02B3 288         .LONG  0      :  
00000000 02B7 289         .LONG  0      :  
02BB 290 :  
02BB 291 : LINE 3A  
02BB 292 :  
00000000 02BB 293 BLKAST: .LONG  0      : Blocking AST (address of .ASCII string)  
00000000 02BF 294 LKB:   .LONG  0      : Address of LKB  
02C3 295 :  
02C3 296 : LINE 4  
02C3 297 :  
00000000 02C3 298 RES2:  .LONG  0      : Second longword of resource name (value)  
00000000 02C7 299 RES1:  .LONG  0      : First longword of resource name (value)  
00000008 02CB 300         .LONG  8      :  
00000000 02CF 301 DMP1:  .LONG  0      : First line of text name (.ASCII string)  
00000000 02D3 302 STS1:  .LONG  0      : 1st line of status (addr. of .ASCII strings)  
00000000 02D7 303         .LONG  0      :  
00000000 02DB 304         .LONG  0      :  
02DF 305 :  
02DF 306 : LINE 5  
02DF 307 :  
00000000 02DF 308 RLEN:  .LONG  0      : Resource name length (value)  
00000000 02E3 309 RES4:  .LONG  0      : Fourth longword of resource name (value)  
00000000 02E7 310 RES3:  .LONG  0      : Third longword of resource name (value)  
00000008 02EB 311         .LONG  8      :  
00000000 02EF 312 DMP3:  .LONG  0      : Second line of text name (.ASCII string)  
00000000 02F3 313 STS2:  .LONG  0      : 2nd line of status (addr. of .ASCII strings)  
00000000 02F7 314         .LONG  0      :  
00000000 02FB 315         .LONG  0      :
```

```
02FF 316
02FF 317 ; LINE 6
02FF 318
00000000 02FF 319 ACMODE: .LONG 0 ; Access mode (Address of .ASCIC string)
00000000 0303 320 RES6: .LONG 0 ; Sixth longword of resource name (value)
00000000 0307 321 RES5: .LONG 0 ; Fifth longword of resource name (value)
00000008 030B 322 .LONG 8
00000000 030F 323 DMP5: .LONG 0 ; Third line of text name (.ASCII string)
00000000 0313 324 STS3: .LONG 0 ; 3rd line of status (addr. of .ASCIC strings)
00000000 0317 325 .LONG 0
00000000 031B 326 .LONG 0
031F 327
031F 328 ; LINE 7
031F 329
00000000 031F 330 NSPACE: .LONG 0 ; Name space (group/system) (addr. of desc.)
00000000 0323 331 RES8: .LONG 0 ; Eighth longword of resource name (value)
00000000 0327 332 RES7: .LONG 0 ; Seventh longword of resource name (value)
00000008 032B 333 .LONG 8
00000000 032F 334 DMP7: .LONG 0 ; Fourth line of text name (.ASCII string)
0333 335
0333 336 ; LINE 8
0333 337
00000000 0333 338 TYPE: .LONG 0 ; Lock type (addr. of .ASCIC string)
00000000 0337 339 RMINFO: .LONG 0 ; Remote info (addr. of .ASCIC string)
033B 340
```



```
00000000 0338 342 .SBTTL READ-ONLY DATA DEFINITIONS
00000000 0000 343 .PSECT LOCK,EXE,NOWRT, LONG
00000000 0000 344
00000000 0000 345 ; HEADERS
00000000 0000 346
00000000 0000 347 LOCK_HEAD: STRING <Lock database>
00015000 0015 348 RES_READ: STRING <Resource database>
0002E000 002E 349
00000000 002E 350 LOCK_STR_TBL: ; table of FAO strings for wait/grant queue display
0000005A 0032 351 .LONG 0
0000006F 0036 352 .LONG LOCKSTR1
00000093 003A 353 .LONG LOCKSTR2
00000093 003A 354 .LONG LOCKSTR3
00000093 003E 355
00000000 003E 356 CONV_STR_TBL: ; table of FAO strings for convert queue display
000000C6 0042 357 .LONG 0
000000DF 0046 358 .LONG CONVSTR1
00000108 004A 359 .LONG CONVSTR2
00000108 004E 360 .LONG CONVSTR3
00000141 004E 361
00000168 0052 362 QUE_STR_TBL: ; table of headers for queue display
0000019B 0056 363 .LONG GRANTSTR
0000019B 0056 364 .LONG CONVSTR
0000019B 005A 365 .LONG WAITSTR
0000019B 005A 366
21 20 20 20 20 20 00000062 010E0000 005A 367 LOCKSTR1: .ASCID # !XL !AC#
43 41 21 20 20 4C 58 0068
21 20 20 20 20 20 00000077 010E0000 006F 368 LOCKSTR2: .ASCID # !XL !AC !XL !AC#
20 20 20 20 20 43 41 21 20 20 4C 58 007D
43 41 21 20 20 4C 58 21 20 20 0089
21 20 20 20 20 20 00000098 010E0000 0093 369 LOCKSTR3: .ASCID # !XL !AC !XL !AC !XL !AC#
20 20 20 20 20 43 41 21 20 20 4C 58 00A1
20 20 43 41 21 20 20 4C 58 21 20 20 00AD
41 21 20 20 4C 58 21 20 20 20 20 20 00B9
43 00C5
21 20 20 20 20 20 000000CE 010E0000 00C6 370
43 41 21 2F 43 41 21 20 20 4C 58 00D4 371 CONVSTR1: .ASCID # !XL !AC/!AC#
21 20 20 20 20 20 000000E7 010E0000 00DF 372 CONVSTR2: .ASCID # !XL !AC/!AC !XL !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58 00ED
2F 43 41 21 20 20 4C 58 21 20 20 20 00F9
43 41 21 0105
21 20 20 20 20 20 00000110 010E0000 0108 373 CONVSTR3: .ASCID # !XL !AC/!AC !XL !AC/!AC !XL !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58 0116
2F 43 41 21 20 20 4C 58 21 20 20 20 0122
20 20 4C 58 21 20 20 20 20 43 41 21 012E
43 41 21 2F 43 41 21 013A
0141
65 74 6E 61 72 47 00000149 010E0000 0141 374
63 6F 4C 28 20 65 75 65 75 71 20 64 014F 375 GRANTSTR: .ASCID #Granted queue (Lock ID / Gr mode):#
6F 6D 20 72 47 20 2F 20 44 49 20 68 015B
3A 29 65 64 0167
72 65 76 6E 6F 43 00000173 010E0000 0168 376 CONVSTR: .ASCID #Conversion queue (Lock ID / Gr/Rq mode):#
28 20 65 75 65 75 71 20 6E 6F 69 73 0179
72 47 20 2F 20 44 49 20 68 63 6F 4C 0185
3A 29 65 64 6F 6D 20 71 52 2F 0191
6E 69 74 69 61 57 000001A3 010E0000 019B 377 WAITSTR: .ASCID #Waiting queue (Lock ID / Rq mode):#
```

LOCK AND RESOURCE FORMATTING ROUTINES K 5
READ-ONLY DATA DEFINITIONS

Page 10
(5)

63	6F	4C	28	20	65	75	65	75	71	20	67	01A9
6F	6D	20	71	52	20	2F	20	44	49	20	68	01B5
								3A	29	65	64	01C1

```

00000000 01C1 378
00000001 01C5 379
00000002 01C5 380
00000003 01C5 381
00000004 01C5 382
00000005 01C5 383
00000006 01C7 384
00000007 01C9 385
00000008 01CB 386
00000009 01CD 387
0000000A 01CF 388

```

Generic Data

LOCKMODE_TBL:

4C 4E	.ASCII	"NL"
52 43	.ASCII	"CR"
57 43	.ASCII	"CW"
52 50	.ASCII	"PR"
57 50	.ASCII	"PW"
58 45	.ASCII	"EX"

```

000001E9' 01D1 390 LKMODE_TBL: .ADDRESS NULL
000001EC' 01D5 391 .ADDRESS CREAD
000001EF' 01D9 392 .ADDRESS CWRITE
000001F2' 01DD 393 .ADDRESS PREAD
000001F5' 01E1 394 .ADDRESS PWRITE
000001F8' 01E5 395 .ADDRESS EX

```

; POINTERS TO MODE TEXT

4C 4E 00'	01E9	396 NULL:	.ASCIC	/NL/
	02 01E9			
52 43 00'	01EC	397 CREAD:	.ASCIC	/CR/
	02 01EC			
57 43 00'	01EF	398 CWRITE:	.ASCIC	/CW/
	02 01EF			
52 50 00'	01F2	399 PREAD:	.ASCIC	/PR/
	02 01F2			
57 50 00'	01F5	400 PWRITE:	.ASCIC	/PW/
	02 01F5			
58 45 00'	01F8	401 EX:	.ASCIC	/EX/
	02 01F8			

```
74 61 20 64 65 74 6E 61 72 47 00' 01FB 403 GR-SIRING:  
01FB 404 .ASCIC "Granted at"
```

```

0208 405 CV_STRING:
20 67 6E 69 74 72 65 76 6E 6F 43 00' 0206 406 .ASCII "Converting to"

```

```
0214 407 WT_STRING:  
0214 408 .ASCII 'Waiting for'
```

```

20 20 0220 409
20 20 0220 410 BLANKS: .ASCII " "

```

```

412 NULL_CSTRING:
413     .ASCII ""

```

54 53 41 4B 4C 42 00' 0223 415 BL-STING:
0223 416 .ASCIC 'BLKAST'

20 4B 4C 42 4C 41 56 00' 022A 418 FLAGS-IBL:
022A 419 .ASCIC "VALBLK "

54 52 45 56 4E 4F 43	00'	07	022A	420	.ASCIC	"CONVERT"
45 55 45 55 51 4F 4E	00'	07	0232	421	.ASCIC	"NOQUEUE"
53 54 53 43 4E 59 53	00'	07	023A	422	.ASCIC	"SYNCSTS"
20 4D 45 54 53 59 53	00'	07	0242	423	.ASCIC	"SYSTEM "
41 54 4F 55 51 4F 4E	00'	07	024A	424	.ASCIC	"NOQUOTA"
20 53 59 53 54 56 43	00'	07	0252	425	.ASCIC	"CVTSYS "
52 45 56 4F 43 45 52	00'	07	025A	426	.ASCIC	"RECOVER"
54 43 45 54 4F 52 50	00'	07	0262	427	.ASCIC	"PROTECT"
		07	026A	428		
		07	0272	429		
54 53 41 4C 50 43 44	00'	07	0272	430	STATUS_TBL:	"DCPLAST"
54 53 41 4B 4C 42 44	00'	07	027A	431	.ASCIC	"DBLKAST"
20 20 43 4E 59 53 41	00'	07	027A	432	.ASCIC	"ASYNCH "
44 51 54 53 41 4C 42	00'	07	0282	433	.ASCIC	"BLASTQD"
20 59 50 43 54 53 4D	00'	07	028A	434	.ASCIC	"MSTCPY "
41 54 4F 55 51 4F 4E	00'	07	0292	435	.ASCIC	"NOQUOTA"
51 54 55 4F 4D 49 54	00'	07	029A	436	.ASCIC	"TIMOUTQ"
20 53 59 53 53 41 57	00'	07	02A2	437	.ASCIC	"WASSYS "
53 59 53 4F 54 56 43	00'	07	02AA	438	.ASCIC	"CVTOSYS"
		07	02B2	439		
6C 61 63 6F 4C	00'	07	02BA	440	LOCAL:	"Local"
	05	07	02BA	441	PROCESS:	
73 73 65 63 6F 72 50	00'	07	02C0	442	.ASCIC	"Process"
72 65 74 73 61 4D	00'	07	02C0	443	MASTER:	"Master"
	06	07	02C8	444		
	20	00'	02CF	445	SPACE:	" "
	01	07	02CF	446		
64 69 6C 61 76 20 74 6F 4E	00'	07	02D1	447	NOT_VALID:	
	09	07	02D1	448	.ASCIC	"Not valid"
65 20 79 72 6F 74 63 65 72 69 44	00'	07	02DB	449	DIR_ENTRY:	
	79 72 74 6E	07	02DB	450	.ASCIC	"Directory entry"
	0F	07	02E7			
		07	02DB	451		
		07	02EB			

LOCK
V04-000

LOCK AND RESOURCE FORMATTING ROUTINES
READ-ONLY DATA DEFINITIONS

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

Page 12
(5)

65 64 6F 6D 20 6C 65 6E 72 65 4B 00'	02EB	452 KMODE: .ASCIC 'Kernel mode'
0B	02EB	
65 64 6F 6D 20 2E 63 65 78 45 00'	02F7	453 EMODE: .ASCIC 'Exec. mode'
0A	02F7	
65 64 6F 6D 20 2E 72 65 70 75 53 00'	0302	454 SMODE: .ASCIC 'Super. mode'
0B	0302	
65 64 6F 6D 20 72 65 73 55 00'	030E	455 UMODE: .ASCIC 'User mode'
09	030E	
	0318	456
	0318	457 .ALIGN LONG
	0318	458 ACMODE_TBL:
000002EB'	0318	459 .ADDRESS KMODE
000002F7'	031C	460 .ADDRESS EMODE
00000302'	0320	461 .ADDRESS SMODE
0000030E'	0324	462 .ADDRESS UMODE
	0328	463

LOC
V04

```
0328 465 .SBTTL SHOW_ALL_LOCKS - ACTION ROUTINE TO DISPLAY ALL LOCKS
0328 466 :+++
0328 467 :
0328 468 SHOW_ALL_LOCKS
0328 469 :
0328 470 PURPOSE
0328 471 ACTION ROUTINE FOR THE "SHOW LOCKS/ALL" COMMAND
0328 472 :
0328 473 INPUT
0328 474 NONE
0328 475 :
0328 476 OUTPUT
0328 477 DISPLAYED DATA
0328 478 :
0328 479 :---
0000 0328 480 .ENTRY SHOW_ALL_LOCKS,*M<>
032A 481 :
6D 000009E8'EF 9E 032A 482 MOVAB LOCK_COND_HAND,(FP) : ESTABLISH CONDITION HANDLER
00000090'EF D4 0331 483 CLRL LOCKID : INITIALIZE VARIABLE
FCC5 CF 9F 0337 484 PUSHAB LOCK_HEAD : GET HEADER
00000000'EF 01 FB 033B 485 CALLS #1,SET_HEADING : SET IT UP
00000000'EF D4 0342 486 CLRL LOCK_COUNT : INIT COUNT OF LOCKS PROCESSED
0348 487 :
0348 488 :
0348 489 STEP THROUGH LOCK ID TABLE, DISPLAYING ENTRIES WITH DATA
0348 490 :
00000090'EF D6 0348 491 10$: INCL LOCKID : INCREMENT INDEX
00000443'EF 00 FB 034E 492 CALLS #0,GET_LKB : GET ADDRESS OF ASSOCIATED LKB
F0 50 E9 0355 493 BLBC R0,10$ : IF LBC, NON-EXISTENT, TRY NEXT
00000000'EF D6 0358 494 INCL LOCK_COUNT : GOT ONE
000004B7'EF 00 FB 035E 495 CALLS #0,SAVE_LOCK_DATA : GET ASSOCIATED DATA
0000051D'EF 00 FB 0365 496 CALLS #0,DISPLAY_LOCK : DISPLAY DATA FOR THIS LOCKID
DA 11 036C 497 BRB 10$ : LOOP (SIGNAL WILL EXIT FOR US)
```

```
036E 499 .SBTTL SHOW_ONE_LOCK - ACTION ROUTINE TO DISPLAY ONE LOCK
036E 500 :+++
036E 501 :
036E 502 SHOW_ONE_LOCK
036E 503 :
036E 504 PURPOSE
036E 505 ACTION ROUTINE TO PROCESS "SHOW LOCK lockid" COMMAND
036E 506 :
036E 507 INPUT
036E 508 LOCKID - INDEX INTO LOCK ID TABLE
036E 509 :
036E 510 OUTPUT
036E 511 DISPLAYED DATA
036E 512 :
036E 513 :---
036E 514 .ENTRY SHOW_ONE_LOCK,^M<>
0370 515
00000443'EF 00 FB 0370 516 CALLS #0,GET_LKB ; GET LOCK BLOCK ADDRESS
21 50 E9 0377 517 BLBC R0,10$ ; IF LBC, NO LOCK BLOCK FOR ID
037A 518
000004B7'EF 00 FB 037A 519 CALLS #0,SAVE_LOCK_DATA ; SAVE LOCK DATA
FC7B CF 9F 0381 520 PUSHAB LOCK_HEAD ; SET UP LOCK HEADING
00000000'EF 01 FB 0385 521 CALLS #1,SET_HEADING
0000051D'EF 00 FB 038C 522 CALLS #0,DISPLAY_LOCK ; DISPLAY LOCK INFO
0393 523 STATUS SUCCESS ; SUCCESSFUL COMPLETION
04 039A 524 RET
039B 525
04 039B 526 10$: SIGNAL 0,NOLKB ; NO LOCK BLOCK FOR THIS ID
03AD 527 RET
```



```
03AE 529 .SBTTL SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
03AE 530
03AE 531
03AE 532
03AE 533
03AE 534
03AE 535
03AE 536
03AE 537
03AE 538
03AE 539
03AE 540
03AE 541
03AE 542
03AE 543
03AE 544
03AE 545
03AE 546
03B0 547
03B0 548
03B4 549
03B9 550
03BD 551
03C2 552
03C5 553
03C7 554
03C7 555
03D3 556
03DA 557
03E1 558
03EE 559
03F5 560
040D 561
0416 562
041D 563
0429 564
042C 565
042E 566
0430 567
0430 568
0430 569
0442 570
04 0442

000C
53 04 AC D0
53 0104 C3 D0
52 08 BC D0
52 0104 C2 DE
52 53 D1
69 13

000002BF'EF 53 00000040 8F C3
000004B7'EF 00 FB
00000556'EF 00 FB
00000856'EF 00 FB
52 53 D1
14 13
97 11

04 0442

529 .SBTTL SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
530
531
532 SHOW_PROC_LOCK
533
534 PURPOSE
535 SHOW LOCKS ASSOCIATED WITH A GIVEN LOCK. CALLED FROM
536 GENERIC CODE WHICH DISPLAYS PROCESS DATA.
537
538 INPUT
539 4(AP) - ADDRESS OF PCB COPY IN SDA IMAGE
540 8(AP) - ADDRESS OF 'REAL' PCB IN SYSTEM
541
542 OUTPUT
543 DISPLAYED DATA
544
545 .ENTRY SHOW_PROC_LOCK, "M<R2,R3>"
546
547 MOVL 4(AP),R3 ; GET ADDRESS OF PCB SAVE AREA
548 MOVL PCB$LOCKQFL(R3),R3 ; GET FORWARD LINK
549 MOVL 8(AP),R2 ; GET 'REAL' ADDRESS OF PCB
550 MOVAL PCB$LOCKQFL(R2),R2 ; GET 'REAL' ADDRESS OF HEAD OF QUEUE
551 CMPL R3,R2 ; ARE THERE ELEMENTS IN THIS QUEUE?
552 BEQL NOLCK ; IF EQL, NO
553
554 GETLCK: SUBL3 #LKB$OWNQFL,R3,LKB ; ADJUST FOR START OF LKB
555 CALLS #0,SAVE_LOCK_DATA ; SAVE LOCK DATA
556 SKIP PAGE
557 PRINT 0,<Lock data:>
558 CALLS #0,FORMAT_LOCK ; FORMAT LOCK DATA
559 ENSURE 9
560 SKIP 1
561 CALLS #0,PRINT_LOCK ; DISPLAY LOCK INFO
562 REQMEM (R3),R3 ; GET NEXT QUEUE LINK
563 CMPL R3,R2 ; AT END OF QUEUE?
564 BEQL DONE ; IF EQL, DONE
565 BRB GETLCK ; LOOP
566
567 NOLCK: ; THIS PROCESS HAS NO LOCKS
568 SIGNAL 0,NOPRLOCK
569
570 DONE: RET ; SUCCESSFUL
```

```
0443 572 .SBTTL GET_LKB - GET LOCK BLOCK
0443 573 :+++
0443 574 :
0443 575 GET_LKB
0443 576 :
0443 577 PURPOSE
0443 578 GIVEN LOCK ID, GET ADDRESS OF LOCK BLOCK.
0443 579 :
0443 580 INPUT
0443 581 LOCKID - INDEX INTO LOCK ID TABLE
0443 582 :
0443 583 OUTPUT
0443 584 R0 - IF LBS, LKB CONTAINS ADDRESS OF LOCK BLOCK
0443 585 IF LBC, NO LOCK BLOCK FOR THIS ID
0443 586 :
0443 587 :---
000C 0443 588 .ENTRY GET_LKB,*M<R2,R3>
0443 589 :
0443 590 :
0443 591 VALIDATE INPUT
0443 592 :
52 00000090'EF 3C 0443 593 MOVZWL LOCKID,R2 ; MOVE TO REGISTER AND TEST
43 13 044C 594 BEQL 20$ ; IF EQL, NOT VALID
044E 595 REQMEM @LCK$GL_MAXID,R3 ; GET MAX ID VALUE
53 52 D1 045E 596 CMPL R2,R3 ; CHECK IT
41 1A 0461 597 BGTRU 30$ ; IF GTRU, TOO BIG
0463 598 :
0463 599 :
0463 600 GET ADDRESS OF LOCK BLOCK
0463 601 :
0463 602 :
0463 603 REQMEM @LCK$GL_IDTBL,R3 ; GET START OF TABLE
53 6342 DE 0473 604 MOVAL (R3)[R2],R3 ; CALC SLOT ADDRESS
0477 605 REQMEM (R3),R3 ; GET LKB ADDRESS
0483 606 CLRL R0 ; ASSUME EMPTY SLOT
000002BF'EF 50 D4 0483 605 MOVL R3,LKB ; MOVE DATA AND TEST
53 D0 0485 606 BGEQ 10$ ; IF GEQ, NO LKB FOR THIS ID
02 18 048C 607 INCL R0 ; INDICATE SUCCESS
50 D6 048E 608 :
0490 609 :
04 0490 610 10$: RET ; DONE
0491 611 :
0491 612 20$: SIGNAL 0,LOCKIDZER ; LOCK ID 0
04 04A3 613 RET
04A4 614 :
04A4 615 30$: SIGNAL 0,OUTOFRANG ; LOCK ID GREATER THAN MAXID
04 04B6 616 RET
```

```
0487 618 .SBTTL SAVE_LOCK_DATA - RETRIEVE LOCK DATA FROM SYSTEM
0487 619 :+++
0487 620 :
0487 621 : SAVE_LOCK_DATA
0487 622 :
0487 623 : PURPOSE
0487 624 : GIVEN ADDRESS OF LOCK BLOCK, RETRIEVE ALL DATA NEEDED TO
0487 625 : DISPLAY LOCK INFORMATION
0487 626 :
0487 627 : INPUT
0487 628 : LKB - ADDRESS OF LOCK BLOCK
0487 629 :
0487 630 : OUTPUT
0487 631 : LKB_BFR - CONTENTS OF LOCK BLOCK
0487 632 : PARID_BFR - ID OF PARENT LOCK
0487 633 : LKB_RSB_BFR - DATA FROM ASSOCIATED RESOURCE
0487 634 :
0487 635 :---
000C 0487 636 .ENTRY SAVE_LOCK_DATA,^M<R2,R3>
0489 637
52 000002BF'EF D0 0489 638 MOVL LKB,R2
04C0 639 REQMEM (R2),LKB_BFR,#LKB$K_LENGTH ; STORE LKB DATA
53 00000050'EF D0 04D5 640 MOVL LKB_BFR+LKB$K_PARENT,R3 ; PARENT LKB ADDR
OD 13 04DC 641 BEQL 10$ ; IF EQL, NO PARENT
04DE 642 REQMEM LKB$K_LKID(R3),R3 ; FETCH PARENT LOCK ID
04EB 643
00000004'EF 53 D0 04EB 644 10$: MOVL R3,PARID_BFR ; STORE IN BUFFER
53 00000058'EF D0 04F2 645 MOVL LKB_BFR+LKB$K_RSB,R3 ; GET ASSOCIATED RSB
04F9 646
04F9 647 ASSUME RSB$B_RMOD-RSB$W_GROUP EQ 2 ; MAKE SURE NOTHING CHANGES
04F9 648 ASSUME RSB$B_RSNLEN-RSB$W_GROUP EQ 3
04F9 649 ASSUME RSB$T_RESNAM-RSB$W_GROUP EQ 4
04F9 650 ASSUME RSB$K_MAXLEN EQ <^X1F>
04F9 651
04F9 652 REQMEM RSB$W_GROUP(R3),LKB_RSB_BFR,- ; STORE RSB DATA
04F9 653 #RSB$K_MAXLEN+4
050B 654 REQMEM RSB$K_CSID(R3),RSB$CSID_BFR ; GET CSID IN RSB
051C 655
04 051C 656 RET
```

```

051D 658 .SBTTL DISPLAY_LOCK - CONTROL DISPLAY OF LOCK DATA
051D 659 :+++
051D 660 :
051D 661 DISPLAY_LOCK
051D 662 :
051D 663 PURPOSE
051D 664 CONTROLS FORMAT AND DISPLAY OF LOCK DATA
051D 665 :
051D 666 INPUT
051D 667 LOCK DATA AREAS (IMPLIED)
051D 668 :
051D 669 OUTPUT
051D 670 DISPLAYED LOCK INFORMATION
051D 671 :
051D 672 :---
0000 051D 673 .ENTRY DISPLAY_LOCK,^M<>
051F 674
00000556'EF 00 FB 051F 675 CALLS #0,FORMAT_LOCK ; FORMAT LOCK DATA
0526 676 SKIP PAGE
052D 677 ENSURE 10
0545 678 SKIP 1
00000856'EF 00 FB 054E 679 CALLS #0,PRINT_LOCK ; DISPLAY LOCK DATA
0555 680
04 0555 681 RET
    
```



```
0556 683 .SBTTL FORMAT_LOCK - FORMAT LOCK DATA
0556 684 :+++
0556 685
0556 686 FORMAT_LOCK
0556 687
0556 688 PURPOSE
0556 689 FORMAT RAW SYSTEM LOCK AND RESOURCE DATA. STORE
0556 690 FOR EASY ACCESS.
0556 691
0556 692 INPUT
0556 693 LKB_BFR - ADDRESS OF LOCK BLOCK DATA
0556 694 PARID_BFR - PARENT LOCK ID
0556 695 LKB_RSB_BFR - ASSOCIATED RESOURCE DATA
0556 696
0556 697 OUTPUT
0556 698 STORED DATA
0556 699
0556 700 :---
0556 701
007C 0556 702 .ENTRY FORMAT_LOCK,^M<R2,R3,R4,R5,R6>
0558 703
0558 704 : Lock id., parent id., sublocks and PID
0558 705
0558 706 MOVAB LKB_BFR,R6
055F 707 MOVL LKB$$_LKID(R6),LKID
0567 708 MOVL PARID_BFR,PARID
0572 709 MOVZWL LKB$_REFCNT(R6),SUBLKS
057A 710 MOVL LKB$_PID(R6),PID
0582 711
0582 712 : Lock state and mode(s)
0582 713
0582 714 TSTB LKB$_STATE(R6)
0585 715 BLSS 20$ : Waiting
0587 716 BGTR 10$ : Granted
0589 717 MOVAB CV_STRING,STATE1
0592 718 MOVAB GR_STRING,STATE2
059B 719 MOVZBL LKB$_GRMODE(R6),R0 : Get granted mode
059F 720 MOVAB LOCKMODE_TBL[R0],MODE2 : Store granted mode sting
05A9 721 MOVZBL LKB$_RQMODE(R6),R0 : Get requested mode
05AD 722 BRB 30$
05AF 723 10$: MOVAB GR_STRING,STATE1
05B8 724 MOVZBL LKB$_GRMODE(R6),R0
05BC 725 BRB 25$
05BE 726 20$: MOVAB WT_STRING,STATE1
05C7 727 MOVZBL LKB$_RQMODE(R6),R0
05CB 728 25$: MOVAB NULL_CSTRING,STATE2 : No state 2
05D4 729 MOVAB BLANKS,MODE2 : No mode 2
05DD 730 30$: MOVAB LOCKMODE_TBL[R0],MODE1 : Store mode 1
05E7 731
05E7 732 : Blocking AST
05E7 733
05E7 734 MOVAB NULL_CSTRING,BLKAST : Assume no blocking AST
05F0 735 TSTL LKB$_BLKASTADR(R6) : Is there a blocking AST?
05F3 736 BEQL 32$
05F5 737 MOVAB BL_STRING,BLKAST : Yes, store address of string
05FE 738
05FE 739 : Flags
```

56 00000008'EF 9E 0558 706
0000026F'EF 30 A6 D0 055F 707
00000283'EF 00000004'EF D0 0567 708
0000029F'EF 4C A6 3C 0572 709
00000273'EF 0C A6 D0 057A 710
36 A6 95 0582 714
37 19 0585 715
26 14 0587 716
00000287'EF FC79 CF 9E 0589 717
000002A3'EF FC65 CF 9E 0592 718
50 35 A6 9A 059B 719
000002AB'EF FC21 CF40 3E 059F 720
50 34 A6 9A 05A9 721
2E 11 05AD 722
00000287'EF FC48 CF 9E 05AF 723 10\$:
50 35 A6 9A 05B8 724
OD 11 05BC 725
00000287'EF FC52 CF 9E 05BE 726 20\$:
50 34 A6 9A 05C7 727
000002A3'EF FC53 CF 9E 05CB 728 25\$:
000002AB'EF FC48 CF 9E 05D4 729
0000028F'EF FBE3 CF40 3E 05DD 730 30\$:
05E7 731
05E7 732
05E7 733
000002BB'EF FC37 CF 9E 05E7 734
20 A6 D5 05F0 735
09 13 05F3 736
000002BB'EF FC2A CF 9E 05F5 737
05FE 738
05FE 739

```

      53 FC20 CF 9E 05FE 740
00000277'EF 53 DO 05FE 741 32$: MOVAB NULL,CSTRING,R3 ; Initialize fields
0000027B'EF 53 DO 0603 742 MOVL R3,FCAGS1
0000027F'EF 53 DO 060A 743 MOVL R3,FLAGS1+4
00000293'EF 53 DO 0611 744 MOVL R3,FLAGS1+8
00000297'EF 53 DO 0618 745 MOVL R3,FLAGS2
0000029B'EF 53 DO 061F 746 MOVL R3,FLAGS2+4
000002AF'EF 53 DO 0626 747 MOVL R3,FLAGS2+8
000002B3'EF 53 DO 062D 748 MOVL R3,FLAGS3
000002B7'EF 53 DO 0634 749 MOVL R3,FLAGS3+4
      55 28 A6 3C 0642 750 MOVL R3,FLAGS3+8
54 00000277'EF 52 DE 0642 751 MOVZWL LKBSW,FLAGS(R6),R5 ; Pick up flags
      55 55 3C 0642 752 MOVZWL LKBSW,FLAGS(R6),R5 ; Pick up flags
      64 FBDO CF 43 7E 0646 753 MOVAL FLAGST,R4 ; Address of 1st arg. list
      54 04 52 D4 064D 754 CLRL R2
      03 52 D4 064F 755 CLRL R3
      54 00000293'EF 09 E1 0651 756 35$: BBC R3,R5,60$ ; Br. if flag not set
      06 52 7E 0655 757 MOVAQ FLAGSTBL[R3],(R4) ; Store appropriate string address
      D3 53 04 C0 065B 758 ADDL #4,R4
      03 52 D6 065E 759 INCL R2 ; Incr. position counter
      54 00000293'EF 09 12 0660 760 CMPL R2,#3 ; Move to 2nd line?
      06 52 D1 0663 761 BNEQ 40$ ; No
      D3 53 09 DE 0665 762 MOVAL FLAGST,R4 ; Yes
      06 52 D1 066C 763 BRB 60$
      54 000002AF'EF 07 12 066E 764 40$: CMPL R2,#6 ; Move to third line
      D3 53 09 DE 0671 765 BNEQ 60$ ; No
      06 52 D1 0673 766 MOVAL FLAGST,R4 ; Yes
      D3 53 09 F2 067A 767 60$: AOBLSS #9,R3,35$ ; Repeat 9 times
      06 52 D1 067E 768
      06 52 D1 067E 769
      06 52 D1 067E 770 PART2:
      06 52 D1 067E 771
      06 52 D1 067E 772 ; Status
      06 52 D1 067E 773
      53 FBA0 CF 9E 067E 774 MOVAB NULL,CSTRING,R3 ; Initialize fields
000002D3'EF 53 DO 0683 775 MOVL R3,STS1
000002D7'EF 53 DO 068A 776 MOVL R3,STS1+4
000002DB'EF 53 DO 0691 777 MOVL R3,STS1+8
000002F3'EF 53 DO 0698 778 MOVL R3,STS2
000002F7'EF 53 DO 069F 779 MOVL R3,STS2+4
000002FB'EF 53 DO 06A6 780 MOVL R3,STS2+8
00000313'EF 53 DO 06AD 781 MOVL R3,STS3
00000317'EF 53 DO 06B4 782 MOVL R3,STS3+4
0000031B'EF 53 DO 06BB 783 MOVL R3,STS3+8
      55 2A A6 3C 06C2 784 MOVZWL LKBSW,STATUS(R6),R5 ; Pick up status
54 000002D3'EF 52 DE 06C2 785 MOVZWL LKBSW,STATUS(R6),R5 ; Pick up status
      55 55 3C 06C2 786 MOVAL STS1,R4 ; Address of 1st arg. list
      64 FB98 CF 43 7E 06C6 787 MOVAL STS1,R4
      54 04 52 D4 06CD 788 CLRL R2
      03 52 D4 06CF 789 CLRL R3
      54 000002F3'EF 09 E1 06D1 789 35$: BBC R3,R5,60$ ; Br. if flag not set
      06 52 7E 06D5 790 MOVAQ STATUS_TBL[R3],(R4) ; Store appropriate string address
      D3 53 04 C0 06DB 791 ADDL #4,R4
      03 52 D6 06DE 792 INCL R2 ; Incr. position counter
      54 000002F3'EF 09 12 06E0 793 CMPL R2,#3 ; Move to 2nd line?
      06 52 D1 06E3 794 BNEQ 40$ ; No
      D3 53 09 DE 06E5 795 MOVAL STS2,R4 ; Yes
      06 52 D1 06EC 796 BRB 60$
```

```

      06 52 D1 06EE 797 40$: CMPL R2,#6 ; Move to third line
      07 12 06F1 798 ; BNEQ 60$ ; No
54 00000313'EF 07 DE 06F3 799 ; MOVAL STS3,R4 ; Yes
      D3 53 09 F2 06FA 800 60$: AOBLS #9,R3,35$ ; Repeat 9 times
      06FE 801 ;
      06FE 802 ; Resource name, length, access mode, and name space
      06FE 803 ;
      06FE 804 ;
      06FE 805 ;
000002DF'EF 0000006F'EF 9A 06FE 806 MOVZBL RSNLEN_BFR,RLEN
      51 0000006F'EF 9A 0709 807 MOVZBL RSNLEN_BFR,R1
      50 00000070'EF 9E 0710 808 MOVAB RESNAM_BFR,R0
      50 DD 0717 809 PUSHL R0 ; Save R0
      60 20 00 60 51 2C 0719 810 MOVCS R1,(R0),#0,#32,(R0) ; Zero out rest of buffer
      50 BED0 071F 811 POPL R0 ; Restore R0
      000002CF'EF 60 DE 0722 812 MOVAL (R0),DMP1
      000002C7'EF 80 DO 0729 813 MOVL (R0)+,RES1
      000002C3'EF 80 DO 0730 814 MOVL (R0)+,RES2
      000002EF'EF 60 DE 0737 815 MOVAL (R0),DMP3
      000002E7'EF 80 DO 073E 816 MOVL (R0)+,RES3
      000002E3'EF 80 DO 0745 817 MOVL (R0)+,RES4
      0000030F'EF 60 DE 074C 818 MOVAL (R0),DMP5
      00000307'EF 80 DO 0753 819 MOVL (R0)+,RES5
      00000303'EF 80 DO 075A 820 MOVL (R0)+,RES6
      0000032F'EF 60 DE 0761 821 MOVAL (R0),DMP7
      00000327'EF 80 DO 0768 822 MOVL (R0)+,RES7
      00000323'EF 80 DO 076F 823 MOVL (R0)+,RES8
      50 0000006E'EF 9A 0776 824 MOVZBL RMOD_BFR,R0 ; Get access mode
000002FF'EF FB96 CF40 DO 077D 825 MOVL ACMODE_TBL[R0],ACMODE
      000001BA'EF B5 0787 826 TSTW GROUP_BUF ; System names have group = 0
      44 13 078D 827 BEQL 70$ ; Branch if system name
      078F 828 SFAQ_S CTRSTR = FAQ GROUP_DSC,-
      078F 829 OUTBUF = GROUP_BUF_DSC,-
      078F 830 P1 = GROUP_BFR
      03 50 E8 07AA 831 BLBS R0,65$
      00A5 31 07AD 832 BRW 90$
000001EB'EF 000001BA'EF B0 07B0 833 65$: MOVW GROUP_BUF,GROUP_NUM
000001EA'EF 000001BC'EF 90 07BB 834 MOVAB GROUP_BUF+2,GROUP_NUM+2
0000031F'EF 000001D8'EF 7E 07C6 835 MOVAQ GROUP_DSC,NSPACE
      0B 11 07D1 836 BRB 80$
0000031F'EF 000001C8'EF 7E 07D3 837 70$: MOVAQ SYS_DSC,NSPACE
      07DE 838 ;
      07DE 839 80$: ; Format type of lock and remote lock info.
      07DE 840 ;
      50 00000068'EF DO 07DE 841 MOVL RSBCSID_BFR,R0 ; Get CSID
      19 12 07E5 842 BNEQ 82$ ; Must be process copy
      1F 2A A6 04 E0 07E7 843 BBS #LKBSV_MSTCPY,LKBSW_STATUS(R6),84$ ; Branch if master copy
00000333'EF FACA CF 9E 07EC 844 MOVAB LOCAL_TYPE ; Set type of lock
00000337'EF FAD6 CF 9E 07F5 845 MOVAB SPACE,RMINFO ; Display no remote info
      52 11 07FE 846 BRB 88$
      0800 847 ;
00000333'EF FABC CF 9E 0800 848 82$: MOVAB PROCESS,TYPE ; Set type of lock
      OD 11 0809 849 BRB 86$
      080B 850 ;
00000333'EF FAB9 CF 9E 080B 851 84$: MOVAB MASTER,TYPE ; Set type of lock
      50 5B A6 DO 0814 852 MOVL LKBSL_(SID(R6),R0 ; Get CSID
      0818 853
```

LOCK
V04-000

J 6
LOCK AND RESOURCE FORMATTING ROUTINES
FORMAT_LOCK - FORMAT LOCK DATA

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

Page 22
(12)

```
51 54 A6 D0 0818 854 86$: MOVL LKBSL REMLKID(R6),R1 ; Get remote lock id.
081C 855 $FAO_5 CTRSTR = FAO RMINFO_DSC,-
081C 856 OUTBUF = RMINFO_DSC,-
081C 857 OUTLEN = RMINFO_LEN,-
081C 858 P1 = R1,-
081C 859 P2 = R0,-
0839 860 BLBC R0,90$
000000C4'EF 19 50 E9 083C 861 MOVB RMINFO_LEN,RMINFO_CNT
00000337'EF 000000B8'EF 90 083C 861 MOVAB RMINFO_CNT,RMINFO- ; Store ptr to .ASCII string
0847 862
0852 863
50 01 D0 0852 864 88$: MOVL #1,R0
04 0855 865 90$: RET
```



```
0856 867 .SBTTL PRINT_LOCK - OUTPUT LOCK DATA
0856 868 :+++
0856 869 :
0856 870 PRINT_LOCK
0856 871 :
0856 872 PURPOSE
0856 873 OUTPUT FORMATTED LOCK DATA, LINE-BY-LINE.
0856 874 :
0856 875 INPUT
0856 876 FORMATTED LOCK DATA
0856 877 :
0856 878 OUTPUT
0856 879 DISPLAYED DATA
0856 880 :
0856 881 :---
0000 0856 882 .ENTRY PRINT_LOCK,*M<>
0858 883 :
0858 884 :
0858 885 LINE 1
0858 886 :
0000027F'EF DD 0858 887 PUSHL FLAGS1+8
0000027B'EF DD 085E 888 PUSHL FLAGS1+4
00000277'EF DD 0864 889 PUSHL FLAGS1
00000273'EF DD 086A 890 PUSHL PID
0000026F'EF DD 0870 891 PUSHL LKID
0876 892 PRINT 5,<Lock id: !XL PID: !XL Flags: !3(8AC)>
0883 893 : LKID PID
0883 894 :
0883 895 :
0883 896 LINE 2
0883 897 :
00000298'EF DD 0883 898 PUSHL FLAGS2+8
00000297'EF DD 0889 899 PUSHL FLAGS2+4
00000293'EF DD 088F 900 PUSHL FLAGS2
0000028F'EF DD 0895 901 PUSHL MODE1
00000287'EF DD 089B 902 PUSHL #2
00000283'EF DD 089D 903 PUSHL STATE1
08A3 904 PUSHL PARID
08A9 905 PRINT 7,<Par. id: !XL !13AC !AD!12* !3(8AC)>
0886 906 : PARID STATE1 MODE1 FLAGS2
0886 907 :
0886 908 :
0886 909 LINE 3
0886 910 :
000002B7'EF DD 0886 911 PUSHL FLAGS3+8
000002B3'EF DD 088C 912 PUSHL FLAGS3+4
000002AF'EF DD 08C2 913 PUSHL FLAGS3
000002AB'EF DD 08C8 914 PUSHL MODE2
000002A3'EF DD 08CE 915 PUSHL #2
0000029F'EF DD 08D0 916 PUSHL STATE2
08D6 917 PUSHL SUBLKS
08DC 918 PRINT 7,<Sublocks: !7UL !13AC !AD!12* !3(8AC)>
08E9 919 : SUBLKS STATE2 MODE2 FLAGS3
08E9 920 :
08E9 921 :
08E9 922 LINE 3A
08E9 923 :
```

```
000002BB'EF DD 08E9 924 PUSHL BLKAST
000002BF'EF DD 08EF 925 PUSHL LKB
                                08F5 926 PRINT 2,<LKB: !XL !9AC>
                                0902 927 : LKB BLKAST
                                0902 928 :
                                0902 929 :
                                0902 930 : LINE 4
                                0902 931 :
000002DB'EF DD 0902 932 PUSHL STS1+8
000002D7'EF DD 0908 933 PUSHL STS1+4
000002D3'EF DD 090E 934 PUSHL STS1
000002CF'EF DD 0914 935 PUSHL DMP1
                                08 DD 091A 936 PUSHL #8
000002C7'EF DD 091C 937 PUSHL RES1
000002C3'EF DD 0922 938 PUSHL RES2
                                0928 939 PRINT 7,<Resource: !XL !XL !AF Status: !3(8AC)>
                                0935 940 : RES2 RES1 DMP1 STS1
                                0935 941 :
                                0935 942 : LINE 5
                                0935 943 :
                                0935 944 :
000002FB'EF DD 0935 945 PUSHL STS2+8
000002F7'EF DD 093B 946 PUSHL STS2+4
000002F3'EF DD 0941 947 PUSHL STS2
000002EF'EF DD 0947 948 PUSHL DMP3
                                08 DD 094D 949 PUSHL #8
000002E7'EF DD 094F 950 PUSHL RES3
000002E3'EF DD 0955 951 PUSHL RES4
000002DF'EF DD 095B 952 PUSHL RLEN
                                0961 953 PRINT 8,< Length !27B !XL !XL !AF !3(8AC)>
                                096E 954 : RLEN RES4 RES3 DMP3 STS2
                                096E 955 :
                                096E 956 : LINE 6
                                096E 957 :
                                096E 958 :
0000031B'EF DD 096E 959 PUSHL STS3+8
00000317'EF DD 0974 960 PUSHL STS3+4
00000313'EF DD 097A 961 PUSHL STS3
0000030F'EF DD 0980 962 PUSHL DMP5
                                08 DD 0986 963 PUSHL #8
00000307'EF DD 0988 964 PUSHL RES5
00000303'EF DD 098E 965 PUSHL RES6
000002FF'EF DD 0994 966 PUSHL ACMODE
                                099A 967 PRINT 8,< !11AC !XL !XL !AF !3(8AC)>
                                09A7 968 : ACMODE RES6 RES5 DMP5 STS3
                                09A7 969 :
                                09A7 970 : LINE 7
                                09A7 971 :
                                09A7 972 :
0000032F'EF DD 09A7 973 PUSHL DMP7
                                08 DD 09AD 974 PUSHL #8
00000327'EF DD 09AF 975 PUSHL RES7
00000323'EF DD 09B5 976 PUSHL RES8
0000031F'EF DD 09BB 977 PUSHL NSPACE
                                09C1 978 PRINT 5,< !11AS !XL !XL !AF>
                                09CE 979 : NSPACE RES8 RES7 DMP7
                                09CE 980 :
```

LOCK
V04-000

LOCK AND RESOURCE FORMATTING ROUTINES^{M 6}
PRINT_LOCK - OUTPUT LOCK DATA

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

Page 25
(13)

```
00000337'EF DD 09CE 981 :  
00000333'EF DD 09CE 982 : LINE 8  
09CE 983 :  
09CE 984 :  
DD 09D4 985 PUSHL RMINFO  
09DA 986 PUSHL TYPE  
09E7 987 PRINT 2,<!AC copy !AC!/>  
04 09E7 988 : TYPE RMINFO  
RET
```

```
09E8 990      .SBTTL LOCK_COND_HAND - CONDITION HANDLER FOR SHOW_ALL_LOCKS
09E8 991      :+++
09E8 992
09E8 993      LOCK_COND_HAND
09E8 994
09E8 995      PURPOSE
09E8 996      PROVIDE EXIT PATH FOR SHOW_ALL_LOCKS WHEN THERE ARE NO MORE LOCKS
09E8 997      TO BE PROCESSED
09E8 998
09E8 999      INPUT
09E8 1000      4(AP) = POINTER TO SIGNAL ARGUMENTS
09E8 1001      8(AP) = POINTER TO MECHANISM ARGUMENTS
09E8 1002
09E8 1003      OUTPUT
09E8 1004      POSSIBLE MODIFICATION OF STATUS; POSSIBLE CHANGE IN FLOW
09E8 1005      OF CONTROL.
09E8 1006
09E8 1007      :---
001C 09E8 1008      .ENTRY LOCK_COND_HAND,^M<R2,R3,R4>
09EA 1009
54    52 04 AC 7D 09EA 1010      MOVQ 4(AP),R2          ; GET ADDRESSES OF ARRAYS
54    54 04 A2 D0 09EE 1011      MOVL CHF$&L_SIG NAME(R2),R4      ; GET CONDITION NAME
00000000'8F D1 09F2 1012      CMPL #SS$&_ONWIND,R4          ; ARE WE UNWINDING?
31 13 09F9 1013      BEQL 20$          ; IF EQL, YES
54    00000000'8F D1 09FB 1014      CMPL #MSG$&_OUTOFRANG,R4      ; DID WE GET THE SIGNAL WE'RE LOOKING FOR?
28 12 0A02 1015      BNEQ 20$          ; IF NEQ, NO - PASS IT ON
0A04 1016
00000000'EF D5 0A04 1017      TSTL LOCK_COUNT          ; DID WE PROCESS LOCKS?
12 12 0A0A 1018      BNEQ 10$          ; IF NEQ, YES
0A0C 1019      SIGNAL 0,NOLOCKS          ; DISPLAY INFO MESSAGE
0A1E 1020
0C A3 50 D0 0A1E 1021 10$:      MOVL R0,CHF$&L_MCH_SAVRO(R3)      ; INDICATE SUCCESS AFTER UNWIND
00000000'GF 7E 7C 0A22 1022      CLRQ -(SP)          ; GO BACK TO ESTABLISHER
02 FB 0A24 1023      CALLS #2,G^SYSSUNWIND          ; UNWIND CALL FRAMES
04 0A2B 1024      RET          ; RETURN TO ESTABLISHER
0A2C 1025
50 0000'8F 3C 0A2C 1026 20$:      MOVZWL #SS$&_RESIGNAL,R0      ; RESIGNAL CONDITION
04 0A31 1027      RET
```



```
0A32 1029 .SBTTL SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY ALL RESOURCES
0A32 1030 :+++
0A32 1031 :
0A32 1032 SHOW_ALL_RES
0A32 1033 :
0A32 1034 PURPOSE
0A32 1035 MAIN ROUTINE TO PROCESS THE "SHOW RES/ALL" COMMAND
0A32 1036 :
0A32 1037 INPUT
0A32 1038 NONE
0A32 1039 :
0A32 1040 OUTPUT
0A32 1041 DISPLAYED DATA
0A32 1042 :
0A32 1043 :---
001C 0A32 1044 .ENTRY SHOW_ALL_RES,^M<R2,R3,R4>
0A34 1045 :
0A34 1046 PUSHAB RES HEAD ; SET UP HEADING
0A38 1047 CALLS #1,SET HEADING
0A3F 1048 CLRL RES_COUNT ; INIT COUNT
0A45 1049 CLRL R4 ; INIT HASH TABLE INDEX
0A47 1050 REQMEM @LCK$GL HTBLCNT,R2 ; GET COUNT OF ELEMENTS IN HASH TABLE
0A57 1051 ASHL R2,#1,HTBL CNT
0A5F 1052 MOVL LCK$GL HASHTBL,R3 ; GET ADDRESS OF START OF HASH TABLE
0A66 1053 REQMEM (R3),R3 ; GET FIRST ELEMENT
0A72 1054 :
0A72 1055 :
0A72 1056 : INDEX THROUGH HASH TABLE, LOOKING FOR HASH CHAINS
0A72 1057 :
0A72 1058 10$:
0A72 1059 MOVAL (R3)[R4],R2 ; GET HEAD OF CHAIN
0A76 1060 REQMEM (R2),RSB ; GET FIRST RSB ADDRESS
0A86 1061 TSTL RSB ; IF ZERO, EMPTY CHAIN
0A8C 1062 BEQL 20$
0A8E 1063 :
0A8E 1064 :
0A8E 1065 : SEQUENCE THROUGH HASH CHAIN
0A8E 1066 :
0A8E 1067 15$:
0A8E 1068 INCL RES_COUNT ; INCREMENT COUNT
0A94 1069 REQMEM @RSB,RSB,BFR,#RSB$K_MAXLEN+RSB$K_LENGTH ;SAVE RSB DATA
0AAD 1070 CALLS #0,SHOW_RSB ; DISPLAY RSB
0AB4 1071 CALLS #0,SHOW_QUEUES ; DISPLAY QUEUES
0ABB 1072 MOVL RSB,BFR,RSB ; GET NEXT RSB
0AC6 1073 BLSS 15$
0AC8 1074 :
0AC8 1075 :
0AC8 1076 : OUTER LOOP CONTROL
0AC8 1077 :
0AC8 1078 20$:
0AC8 1079 AOBLS HTBL_CNT,R4,10$ ; INCREMENT INDEX, LOOP IF LSS
0AD0 1080 :
0AD0 1081 TSTL RES_COUNT ; DID WE PROCESS ANY RESOURCES?
0AD6 1082 BEQL 30$ ; IF EQL, NO
0AD8 1083 :
0AD8 1084 STATUS SUCCESS ; SUCCESS RETURN
0ADF 1085 RET
```

00000000'EF 01 FB 0A38 1047
00000115'EF D4 0A3F 1048
54 D4 0A45 1049
00000190'EF 01 52 78 0A57 1051
53 00000000'EF D0 0A5F 1052
00000115'EF D6 0A8E 1065
00000B62'EF 00 FB 0AAD 1070
00000B92'EF 00 FB 0AB4 1071
000001EB'EF 0000011D'EF D0 0ABB 1072
C6 19 0AC6 1073
A2 54 00000190'EF F2 0AC8 1078
00000115'EF D5 0AD0 1080
08 13 0AD6 1082
04 0AD8 1084
0ADF 1085

LOCK AND RESOURCE FORMATTING ROUTINES 16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY 5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAN;1

Page 28
(15)

04 0AEO 1086 30\$: SIGNAL 0,NORESOURC ; NO RESOURCES DISPLAYED
 0AEO 1087 RET
 0AF2 1088

[illegible]

```
0AF3 1090
0AF3 1091
0AF3 1092 :+++
0AF3 1093
0AF3 1094 SHOW_ONE_RES
0AF3 1095
0AF3 1096 PURPOSE
0AF3 1097 MAIN ROUTINE TO PROCESS THE "SHOW RES/LOCK=lockid" COMMAND.
0AF3 1098
0AF3 1099 INPUT
0AF3 1100 LOCKID - ID OF LOCK WHOSE ASSOCIATED RESOURCE IS TO BE DISPLAYED.
0AF3 1101
0AF3 1102 OUTPUT
0AF3 1103 DISPLAYED DATA.
0AF3 1104
0AF3 1105 :---
0004 0AF3 1106 .ENTRY SHOW_ONE_RES,^M<R2>
0AF3 1107
F949 CF 00 FB 0AF3 1108 CALLS #0,GET_LKB ; GET LKB ADDRESS
52 50 E9 0AFA 1109 BLBC R0,10$ ; IF LBC, NONE EXISTS
52 000002BF'EF D0 0AFD 1110 MOVL LKB,R2
0B04 1111 REQMEM LKB$L,RSB(R2),RSB ; GET ADDRESS OF RSB
0B15 1112 REQMEM @RSB,RSB_BFR,#RSB$K_MAXLEN+RSB$K_LENGTH ; GET RSB DATA
F4E3 CF 9F 0B2E 1113 PUSHAB RES_HEAD ; SET RSB HEADING
00000000'EF 01 FB 0B32 1114 CALLS #1,SET_HEADING
00000B62'EF 00 FB 0B39 1115 CALLS #0,SHOW_RSB ; DISPLAY RSB
00000B92'EF 00 FB 0B40 1116 CALLS #0,SHOW_QUEUES ; DISPLAY QUEUES
0B47 1117
0B47 1118 STATUS SUCCESS
04 0B4E 1119 RET
0B4F 1120
04 0B4F 1121 10$: SIGNAL 0,NOLKB ; LOCKID NON-EXISTANT
04 0B61 1122 RET
```

```

00000BCC'EF  00  FB  0000  0B62  1124  .SBTTL  SHOW_RSB - CONTROL FOR RSB DISPLAY
                                0B62  1125  :+++
                                0B62  1126  :
                                0B62  1127  :      SHOW_RSB
                                0B62  1128  :
                                0B62  1129  :      PURPOSE
                                0B62  1130  :      CONTROL THE FORMAT AND DISPLAY OF RSB DATA
                                0B62  1131  :
                                0B62  1132  :      INPUT
                                0B62  1133  :      RSB DATA STRUCTURES (IMPLIED)
                                0B62  1134  :
                                0B62  1135  :      OUTPUT
                                0B62  1136  :      DISPLAYED DATA
                                0B62  1137  :---
00000D34'EF  00  FB  0000  0B62  1138  .ENTRY  SHOW_RSB,^M<>
                                0B64  1139
                                0B64  1140  CALLS  #0,FORMAT_RSB
                                0B68  1141  SKIP   PAGE
                                0B72  1142  ENSURE 9
                                0B8A  1143  CALLS  #0,PRINT_RSB
                                0B91  1144
                                04  0B91  1145  RET

```



```
0B92 1147 .SBTTL SHOW_QUEUES - DISPLAY QUEUES FOR GIVEN RESOURCE
0B92 1148 :+++
0B92 1149 :
0B92 1150 SHOW_QUEUES
0B92 1151 :
0B92 1152 PURPOSE
0B92 1153 DISPLAY, SEQUENTIALLY, THE GRANTED, CONVERSION AND WAIT
0B92 1154 QUEUES ASSOCIATED WITH A RESOURCE.
0B92 1155 :
0B92 1156 INPUT
0B92 1157 RSB - BUFFER CONTAINING RSB DATA
0B92 1158 :
0B92 1159 OUTPUT
0B92 1160 DISPLAYED QUEUES.
0B92 1161 :
0B92 1162 :---
0000 0B92 1163 .ENTRY SHOW_QUEUES,^M<>
0B94 1164 :
0B94 1165 :
7E 10 000001EB'EF C1 0B94 1166 ADDL3 RSB,#RSB$$_GRQFL,-(SP) ; GET ADDRESS OF GRANT QUE FLINK
      FFFFFFFF 8F DD 0B9C 1167 PUSHL #GRANT ; IND GRANT QUE
      00000E55'EF 02 FB 0BA2 1168 CALLS #2,PROCESS_QUEUE
      0BA9 1169 :
7E 18 000001EB'EF C1 0BA9 1170 ADDL3 RSB,#RSB$$_CVTQFL,-(SP) ; GET ADDRESS OF CONVERT QUE FLINK
      00 DD 0BB1 1171 PUSHL #CONVERT ; IND CONVERT QUE
      00000E55'EF 02 FB 0BB3 1172 CALLS #2,PROCESS_QUEUE
      0BBA 1173 :
7E 20 000001EB'EF C1 0BBA 1174 ADDL3 RSB,#RSB$$_WTQFL,-(SP) ; GET ADDRESS OF WAIT QUE FLINK
      01 DD 0BC2 1175 PUSHL #WAIT ; IND WAIT QUE
      00000E55'EF 02 FB 0BC4 1176 CALLS #2,PROCESS_QUEUE
      0BCB 1177 :
      04 0BCB 1178 RET
```

```
OBCC 1180 .SBTTL FORMAT_RSB - FORMAT RSB DATA
OBCC 1181
OBCC 1182
OBCC 1183
OBCC 1184
OBCC 1185
OBCC 1186
OBCC 1187
OBCC 1188
OBCC 1189
OBCC 1190
OBCC 1191
OBCC 1192
OBCC 1193
OBCC 1194
OBCC 1195
OBCC 1196
OBCC 1197
OBCC 1198
OBCC 1199
OBCC 1200
OBCC 1201
OBCC 1202
OBCC 1203
OBCC 1204
OBCC 1205
OBCC 1206
OBCC 1207
OBCC 1208
OBCC 1209
OBCC 1210
OBCC 1211
OBCC 1212
OBCC 1213
OBCC 1214
OBCC 1215
OBCC 1216
OBCC 1217
OBCC 1218
OBCC 1219
OBCC 1220
OBCC 1221
OBCC 1222
OBCC 1223
OBCC 1224
OBCC 1225
OBCC 1226
OBCC 1227
OBCC 1228
OBCC 1229
OBCC 1230
OBCC 1231
OBCC 1232
OBCC 1233
OBCC 1234
OBCC 1235
OBCC 1236

00FC 1195 .ENTRY FORMAT_RSB,^M<R2,R3,R4,R5,R6,R7>
OBCE 1196
OBCE 1197
OBCE 1198
OBCE 1199
OBCE 1200
OBCE 1201
OBCE 1202
OBCE 1203
OBCE 1204
OBCE 1205
OBCE 1206
OBCE 1207
OBCE 1208
OBCE 1209
OBCE 1210
OBCE 1211
OBCE 1212
OBCE 1213
OBCE 1214
OBCE 1215
OBCE 1216
OBCE 1217
OBCE 1218
OBCE 1219
OBCE 1220
OBCE 1221
OBCE 1222
OBCE 1223
OBCE 1224
OBCE 1225
OBCE 1226
OBCE 1227
OBCE 1228
OBCE 1229
OBCE 1230
OBCE 1231
OBCE 1232
OBCE 1233
OBCE 1234
OBCE 1235
OBCE 1236

56 0000011D'EF 9E OBCE 1197 MOVAB RSB_BFR,R6 ; GET ADDRESS OF DATA BUFFER
OBCE 1198
OBCE 1199
OBCE 1200
OBCE 1201
OBCE 1202
OBCE 1203
OBCE 1204
OBCE 1205
OBCE 1206
OBCE 1207
OBCE 1208
OBCE 1209
OBCE 1210
OBCE 1211
OBCE 1212
OBCE 1213
OBCE 1214
OBCE 1215
OBCE 1216
OBCE 1217
OBCE 1218
OBCE 1219
OBCE 1220
OBCE 1221
OBCE 1222
OBCE 1223
OBCE 1224
OBCE 1225
OBCE 1226
OBCE 1227
OBCE 1228
OBCE 1229
OBCE 1230
OBCE 1231
OBCE 1232
OBCE 1233
OBCE 1234
OBCE 1235
OBCE 1236

67 20 00 67 50 A6 9E OBCE 1197 MOVAB RSB_BFR,R6 ; GET ADDRESS OF DATA BUFFER
20 00 67 50 A6 9A OBCE 1198
00000227'EF 67 DE OBE3 1205 MOVAB RSB$T_RESNAM(R6),R7 ; LET R7 POINT TO RESOURCE NAME
0000021F'EF 87 DO OBEA 1206 MOVZBL RSB$B_RSNLEN(R6),R0 ; ZERO OUT UNUSED PORTION OF BUFFER
0000021B'EF 87 DO OBF1 1207 MOVCS R0,(R7),#0,#32,(R7)
0000023B'EF 67 DE OBF8 1208 MOVAL (R7),BUF1
00000233'EF 87 DO OBF8 1209 MOVL (R7)+,RESN1
0000022F'EF 87 DO OC06 1210 MOVL (R7)+,RESN2
00000253'EF 67 DE OC0D 1211 MOVL (R7)+,RESN3
0000024B'EF 87 DO OC14 1212 MOVL (R7)+,RESN4
00000247'EF 87 DO OC1B 1213 MOVL (R7)+,RESN5
0000026B'EF 67 DE OC22 1214 MOVL (R7)+,RESN6
00000263'EF 87 DO OC29 1215 MOVL (R7)+,RESN7
0000025F'EF 87 DO OC30 1216 MOVL (R7)+,RESN8
OC37 1217
OC37 1218
OC37 1219
OC37 1220
OC37 1221
OC37 1222
OC37 1223
OC37 1224
OC37 1225
OC37 1226
OC37 1227
OC37 1228
OC37 1229
OC37 1230
OC37 1231
OC37 1232
OC37 1233
OC37 1234
OC37 1235
OC37 1236

000001F3'EF 48 A6 DO OC37 1221 MOVL RSB$P_PARENT(R6),PRSB
000001FB'EF 40 A6 3C OC3F 1222 MOVZWL RSB$W_REFCNT(R6),RSBCT
000001FF'EF 42 A6 3C OC47 1223 MOVZWL RSB$W_BLKASTCNT(R6),BAST
0000020F'EF 28 A6 DO OC4F 1224 MOVL RSB$Q_VALBLK(R6),VAL1
0000020B'EF 2C A6 DO OC57 1225 MOVL RSB$Q_VALBLK+4(R6),VAL2
00000207'EF 30 A6 DO OC5F 1226 MOVL RSB$Q_VALBLK+8(R6),VAL3
00000203'EF 34 A6 DO OC67 1227 MOVL RSB$Q_VALBLK+12(R6),VAL4
00000213'EF 3C A6 DO OC6F 1228 MOVL RSB$P_VALSEQNUM(R6),SEQNUM
0000023F'EF 38 A6 DO OC77 1229 MOVL RSB$P_CSID(R6),CSID
00000217'EF F64C CF 9E OC7F 1230 MOVAB SPACE,VALID
09 0E A6 01 E1 OC88 1231 BBC #RSB$V_VALINVLID,RSB$W_STATUS(R6),15$
00000217'EF F640 CF 9E OC8D 1232 MOVAB NOT_VALID,VALID
00000257'EF F635 CF 9E OC96 1233 15$: MOVAB SPACE,DIRENTRY
09 0E A6 00 E1 OC9F 1234 BBC #RSB$V_DIRENTRY,RSB$W_STATUS(R6),17$
00000257'EF F633 CF 9E OCA4 1235 17$: MOVAB DIR_ENTRY,DIRENTRY
51 0C A6 9A OCAD 1236 MOVZBL RSB$B_GGMODE(R6),R1
```

LOCK
V04-000

H 7
LOCK AND RESOURCE FORMATTING ROUTINES
FORMAT_RSB - FORMAT RSB DATA

16-SEP-1984 01:31:38
5-SEP-1984 03:32:46

VAX/VMS Macro V04-00
[SDA.SRC]LOCK.MAR;1

Page 33
(19)

```
000001EF'EF  F51B CF41  D0  OCB1 1237  MOVL  LKMODE_TBL[R1],GGMOD
                51  0D A6  9A  OCBB 1238  MOVZBL RSB$B,CGMODE(R6),R1
000001F7'EF  F50D CF41  D0  OCBF 1239  MOVL  LKMODE_TBL[R1],CGMOD
                51  4E A6  90  OCC9 1240  MOVB  RSB$B,RMOD(R6),R1
00000243'EF  F646 CF41  D0  OCCD 1241  MOVL  ACMODE_TBL[R1],RACMOD
                0000022B'EF  4F A6  9A  OCD7 1242  MOVZBL RSB$B,RSNLEN(R6),RESNLEN
0000025B'EF  000001C8'EF  7E  OCDF 1243  MOVAQ  SYS_DSC,RNSPACE
                0000006C'EF  4C A6  B0  OCEA 1244  MOVW  RSB$W,GROUP(R6),GROUP_BFR
                3F  13  OCF2 1245  BEQL  90$
                OCF4 1246  $FAO_5
                OCF4 1247  CTRSTR = FAO_GROUP_DSC,-
                OCF4 1248  OUTBUF = GROUP_BUF_DSC,-
                OD0F 1249  P1 = GROUP_BFR
000001E8'EF  000001BA'EF  E9  OD0F 1249  BLBC  R0,90$
000001EA'EF  000001BC'EF  B0  OD12 1250  MOVW  GROUP_BUF,GROUP_NUM
0000025B'EF  000001D8'EF  90  OD1D 1251  MOVB  GROUP_BUF+2,GROUP_NUM+2
                7E  OD28 1252  MOVAQ  GROUP_DSC,RNSPACE
                OD33 1253
                OD33 1254 90$:
04  OD33 1255  RET
```

: Branch if system name

```
0034 1257 .SBTTL PRINT_RSB - OUTPUT RSB DATA
0034 1258 :+++
0034 1259 :
0034 1260 PRINT_RSB
0034 1261 :
0034 1262 PURPOSE
0034 1263 OUTPUT RSB DATA, LINE-BY-LINE.
0034 1264 :
0034 1265 INPUT
0034 1266 FORMATTED DATA.
0034 1267 :
0034 1268 OUTPUT
0034 1269 DISPLAYED DATA.
0034 1270 :
0034 1271 :---
0000 0034 1272 .ENTRY PRINT_RSB,*M<>
0036 1273 :
0036 1274 :
0036 1275 :
0036 1276 LINE 1
0036 1277 :
000001EF'EF DD 0036 1278 PUSHL GGMOD
000001EB'EF DD 003C 1279 PUSHL RSB
0042 1280 PRINT 2,<Address of RSB: !XL Group grant mode: !AC >
004F 1281 : RSB GGMOD
004F 1282 :
004F 1283 :
004F 1284 LINE 2
004F 1285 :
000001F7'EF DD 004F 1286 PUSHL CGMOD
000001F3'EF DD 0055 1287 PUSHL PRSB
005B 1288 PRINT 2,<Parent RSB: !XL Conversion grant mode: !AC >
0068 1289 : PRSB CGMOD
0068 1290 :
0068 1291 :
0068 1292 LINE 3
0068 1293 :
000001FF'EF DD 0068 1294 PUSHL BAST
000001FB'EF DD 006E 1295 PUSHL SRSBCT
0074 1296 PRINT 2,<Sub-RSB count: !8UL BLKAST count: !8UL>
0081 1297 : SRSBCT BAST
0081 1298 :
0081 1299 :
0081 1300 LINE 4
0081 1301 :
00000217'EF DD 0081 1302 PUSHL VALID
00000213'EF DD 0087 1303 PUSHL SEQNUM
0000020F'EF DD 008D 1304 PUSHL VAL1
00000208'EF DD 0093 1305 PUSHL VAL2
00000207'EF DD 0099 1306 PUSHL VAL3
00000203'EF DD 009F 1307 PUSHL VAL4
00A5 1308 PRINT 6,<Value block: !XL !XL !XL !XL Seq. #: !XL !AC>
00B2 1309 : VAL4 VAL3 VAL2 VAL1 SEQNUM VALID
00B2 1310 :
00B2 1311 :
00B2 1312 LINE 5
00B2 1313 :
```



```
00000227'EF DD ODB2 1314 PUSHL BUF1
0000021F'EF DD ODB8 1315 PUSHL #8
0000021B'EF DD ODBA 1316 PUSHL RESN1
DD ODC0 1317 PUSHL RESN2
DD ODC6 1318 PRINT 4,<Resource: !XL !XL !AF >
DD ODD3 1319 RESN2 RESN1 BUF1
DD ODD3 1320
DD ODD3 1321
DD ODD3 1322
DD ODD3 1323
DD ODD3 1324
DD ODD9 1325
DD ODDF 1326
DD ODE1 1327
DD ODE7 1328
DD ODED 1329
DD ODF3 1330
OE00 1331
OE00 1332
OE00 1333
OE00 1334
OE00 1335
00000257'EF DD OE00 1336 PUSHL DIRENTRY
00000253'EF DD OE06 1337 PUSHL BUF3
0000024B'EF DD OE0C 1338 PUSHL #8
00000247'EF DD OE0E 1339 PUSHL RESN5
00000243'EF DD OE14 1340 PUSHL RESN6
DD OE1A 1341 PUSHL RACMOD
DD OE20 1342 PRINT 6,< !11AC !XL !XL !AF !AC>
DD OE2D 1343 RACMOD RESN6 RESN5 BUF3 DIRENTRY
DD OE2D 1344
DD OE2D 1345
DD OE2D 1346
DD OE2D 1347
0000026B'EF DD OE2D 1348 PUSHL BUF4
00000263'EF DD OE33 1349 PUSHL #8
0000025F'EF DD OE35 1350 PUSHL RESN7
0000025B'EF DD OE3B 1351 PUSHL RESN8
DD OE41 1352 PUSHL RNSPACE
DD OE47 1353 PRINT 5,< !11AS !XL !XL !AF>
DD OE54 1354 RNSPACE RESN8 RESN7 BUF4
DD OE54 1355
04 OE54 1356 RET
```

```
0E55 1358 .SBTTL PROCESS_QUEUE - TRAVERSE RESOURCE QUEUES
0E56 1359 :+++
0E57 1360 :
0E58 1361 :   PROCESS_QUEUE
0E59 1362 :
0E60 1363 :   PURPOSE
0E61 1364 :   TRAVERSE AN INDIVIDUAL RESOURCE QUEUE, FORMATTING AND
0E62 1365 :   DISPLAYING ITS CONTENTS.
0E63 1366 :
0E64 1367 :   INPUT
0E65 1368 :   4(AP) - QUEUE TYPE, WHERE:  -1: GRANT
0E66 1369 :                               0: CONVERSION
0E67 1370 :                               1: WAIT
0E68 1371 :   8(AP) - ADDRESS OF HEAD OF QUEUE
0E69 1372 :
0E70 1373 :   OUTPUT
0E71 1374 :   DISPLAYED CONTENTS OF QUEUE
0E72 1375 :
0E73 1376 :---
0E74 1377 .ENTRY PROCESS_QUEUE, *M<R2,R3,R4,R5,R6,R7,R8,R9>
0E75 1378
0E76 1379 MOVAB LKID_BFR,R4 : GET ADDRESS OF LOCKID STORAGE BUFFER
0E77 1380 MOVAB GRMD_BFR,R5 : GET ADDRESS OF GRANT MODE BUFFER
0E78 1381 MOVAB RQMD_BFR,R6 : GET ADDRESS OF REQUEST MODE BUFFER
0E79 1382 MOVL 8(AP),R2 : GET ADDRESS OF HEAD OF QUEUE
0E80 1383 MOVL R2,R9
0E81 1384 CLRL QUEUE_COUNT : INIT COUNT OF QUEUE ITEMS PROCESSED
0E82 1385 MOVL 4(AP),QUEUE_TYPE : GET TYPE OF QUEUE
0E83 1386 MNEGL #1,R8
0E84 1387 ADDL3 #1,QUEUE_TYPE,R7
0E85 1388 MOVL QUE_STR_TBL[R7],R7 : GET HEADER FOR THIS TYPE OF QUEUE
0E86 1389 ENSURE 3
0E87 1390 SKIP 1
0E88 1391 PRINTD R8,(R7) : DISPLAY IT
0E89 1392 :
0E90 1393 :
0E91 1394 :   INIT FOR SCAN OF QUEUE
0E92 1395 :
0E93 1396 :10$: CLRL R3
0E94 1397 :
0E95 1398 :   GET QUEUE ELEMENT
0E96 1399 :
0E97 1400 :
0E98 1401 :20$: REQMEM (R2),R2 : GET QUEUE ELEMENT
0E99 1402 :      CMPL R9,R2 : AT END OF QUEUE?
0E100 1403 :      BEQL 30$ : IF EQL, YES
0E101 1404 :
0E102 1405 :      ASSUME LKBSL_SQFL-LKBSL_LKID GT 0
0E103 1406 :      ASSUME LKBSB_RQMODE-LKBSL_LKID EQ 4
0E104 1407 :      ASSUME LKBSB_GRMODE-LKBSL_LKID EQ 5
0E105 1408 :
0E106 1409 :      INCL QUEUE_COUNT : INCREMENT COUNT
0E107 1410 :      SUBL3 #<LKBSL_SQFL-LKBSL_LKID>,R2,R8 : GET START OF RELEVANT DATA
0E108 1411 :      REQMEM (R8),LKB_BFR,#6 : SAVE ID, GRANT & REQUEST MODE
0E109 1412 :      MOVAB LKB_BFR,R7
0E110 1413 :      MOVL LKID_OFF(R7),(R4)[R3] : MOVE INTO STORAGE BUFFERS
0E111 1414 :      MOVB GRMD_OFF(R7),(R5)[R3]
```

03FC

54 000001A2'EF 9E 0E57 1379

55 0000019C'EF 9E 0E5E 1380

56 0000019F'EF 9E 0E65 1381

52 08 AC DO 0E6C 1382

59 52 DO 0E70 1383

00000119'EF D4 0E73 1384

00000198'EF 04 AC DO 0E79 1385

58 01 CE 0E81 1386

57 00000198'EF 01 C1 0E84 1387

57 F1BD CF47 DO 0E8C 1388

0E92 1389

0EAA 1390

0EB3 1391

0EBE 1392

0E9E 1393

0EBE 1394

0EBE 1395

53 D4 0EBE 1396

0ECO 1397

0ECO 1398

0ECO 1399

0ECO 1400

0ECO 1401

52 59 D1 0ECC 1402

3D 13 0ECF 1403

0ED1 1404

0ED1 1405

0ED1 1406

0ED1 1407

0ED1 1408

00000119'EF D6 0ED1 1409

58 52 08 C3 0ED7 1410

0EDB 1411

57 00000008'EF 9E 0EEC 1412

6443 67 DO 0EF3 1413

6543 05 A7 90 0EF7 1414

```
6643 04 A7 90 0EFC 1415      MOVB    RQMD_OFF(R7),(R6)[R3]
          0F01 1416
          BB 53 03 F2 0F01 1417      AOBLS    #3,R3,20$      ; IF WE'VE SAVED 3 ITEMS, PRINT THEM
00000F2B'EF 00 FB 0F05 1418      CALLS    #0,PRINT_LINE
          B0 11 0F0C 1419      BRB      10$      ; GET MORE ITEMS
          0F0E 1420
00000F2B'EF 00 FB 0F0E 1421 30$: CALLS    #0,PRINT_LINE      ; PRINT ANY REMAINING ITEMS
          00000119'EF D5 0F15 1422      TSTL    QUEUE_COUNT      ; WERE ANY ITEMS PROCESSED?
          OD 12 0F1B 1423      BNEQ    EXIT      ; IF NEQ, YES
          0F1D 1424      PRINT    0,<      *** EMPTY QUEUE ***>
          0F2A 1425
          04 0F2A 1426 EXIT: RET
```

```
0F2B 1428 .SBTTL PRINT_LINE - OUTPUT QUEUE DATA
0F2B 1429 :+++
0F2B 1430 :
0F2B 1431 PRINT_LINE
0F2B 1432 :
0F2B 1433 PURPOSE
0F2B 1434 GIVEN LOCK DATA FOR ONE LINE (0 - 3 ITEMS), FORMAT AND
0F2B 1435 DISPLAY THIS DATA
0F2B 1436 :
0F2B 1437 INPUT
0F2B 1438 R3 - COUNT OF ITEMS TO DISPLAY
0F2B 1439 R4 - ADDRESS OF BUFFER CONTAINING LOCK IDS
0F2B 1440 R5 - ADDRESS OF BUFFER CONTAINING GRANT-MODES
0F2B 1441 R6 - ADDRESS OF BUFFER CONTAINING REQUESTED-MODES
0F2B 1442 QUEUE_TYPE - TYPE OF QUEUE WE ARE WORKING ON
0F2B 1443 :
0F2B 1444 OUTPUT
0F2B 1445 ONE DISPLAYED LINE OF QUEUE LOCK DATA.
0F2B 1446 :
0F2B 1447 :---
0180 0F2B 1448 .ENTRY PRINT_LINE,*M<R7,R8>
0F2D 1449 :
0F2D 1450 :
0F2D 1451 INITIALIZE
0F2D 1452 :
0F2D 1453 :
57 00000198'EF D0 0F2D 1454 MOVL QUEUE_TYPE,R7 ; SAVE QUEUE TYPE
00000194'EF 53 D0 0F34 1455 MOVL R3,COUNT ; SAVE COUNT
01 53 F4 0F3B 1456 SOBGEQ R3,10$ ; CONVERT TO ARRAY INDEX
04 0F3E 1457 RET ; IF COUNT 0, NO ITEMS TO DISPLAY
0F3F 1458 :
0F3F 1459 :
0F3F 1460 FORMAT DATA
0F3F 1461 :
57 D5 0F3F 1462 10$: TSTL R7 ; ONLY FORMAT RQ MODE IF WAIT OR CONVERT
09 19 0F41 1463 BLSS 15$ ; QUEUE
58 6643 9A 0F43 1464 MOVZBL (R6)[R3],R8
F285 CF48 DD 0F47 1465 PUSHL LKMODE_TBL[R8]
0F4C 1466 :
57 D5 0F4C 1467 15$: TSTL R7 ; ONLY FORMAT GR MODE IF GRANT OR CONVERT
09 14 0F4E 1468 BGTR 20$ ; QUEUE
58 6543 9A 0F50 1469 MOVZBL (R5)[R3],R8
F278 CF48 DD 0F54 1470 PUSHL LKMODE_TBL[R8]
0F59 1471 :
6443 DD 0F59 1472 20$: PUSHL (R4)[R3] ; PROCESS LOCK ID
E0 53 F4 0F5C 1473 SOBGEQ R3,10$ ; IF NOT YET DONE, LOOP
0F5F 1474 :
58 F0DB CF 9E 0F5F 1475 MOVAB CONV_STR_TBL,R8 ; GET ADDRESS OF CORRECT FAQ STRING TABLE
57 D5 0F64 1476 TSTL R7
05 13 0F66 1477 BEQL 25$ ; IF EQL, CONVERT
58 F0C2 CF 9E 0F68 1478 MOVAB LOCK_STR_TBL,R8 ; ELSE WAIT/GRANT
0F6D 1479 :
53 00000194'EF D0 0F6D 1480 25$: MOVL COUNT,R3 ; PICK UP CORRECT STRING FOR NUMBER OF
58 6843 D0 0F74 1481 MOVL (R8)[R3],R8 ; ITEMS
53 53 C0 0F78 1482 ADDL R3,R3 ; CALCULATE CORRECT FAQ ARG COUNT
57 D5 0F7B 1483 TSTL R7
07 12 0F7D 1484 BNEQ 30$
```


LOCK
V04-000

LOCK AND RESOURCE FORMATTING ROUTINES
PRINT_LINE - OUTPUT QUEUE DATA

N 7

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1

Page 39
(22)

```
53 00000194'EF  C0  OF7F 1485      ADDL  COUNT,R3
                   OF86 1486
                   OF86 1487 30$:  PRINTD r3,(r8)      ; DISPLAY LINE
                   OF91 1488
                   04  OF91 1489      RET
                   OF92 1490
                   OF92 1491      .END
```

LOCK
Symbol table

LOCK AND RESOURCE FORMATTING ROUTINES

B 8

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [SDA.SRC]LOCK.MAR;1Page 40
(22)

\$\$T2	= 00000004		GROUP TXT	000001E0	R	02
ACMODE	000002FF	R	02	000001FB	R	03
ACMODE_TBL	00000318	R	03	00000190	R	02
ARGS	= 00000003		HTBL_CNT	0000018C	R	02
BAST	000001FF	R	02	000002EB	R	03
BLANKS	00000220	R	03	*****	X	03
BLKAST	000002BB	R	02	*****	X	03
BL_STRING	00000223	R	03	*****	X	03
BUF1	00000227	R	02	*****	X	03
BUF2	0000023B	R	02	*****	X	03
BUF3	00000253	R	02	*****	X	03
BUF4	0000026B	R	02	*****	X	03
CGMOD	000001F7	R	02	000002BF	R	02
CHFSL_MCH_SAVRO	= 0000000C		LKBSB_GRMODE	= 00000035		
CHFSL_SIG_NAME	= 00000004		LKBSB_RQMODE	= 00000034		
CONVERT	= 00000000		LKBSB_STATE	= 00000036		
CONVSTR	0000016B	R	03	LKBSK_LENGTH	= 00000060	
CONVSTR1	000000C6	R	03	LKBSL_BLKASTADR	= 00000020	
CONVSTR2	000000DF	R	03	LKBSL_CSID	= 00000058	
CONVSTR3	00000108	R	03	LKBSL_LKID	= 00000030	
CONV_STR_TBL	0000003E	R	03	LKBSL_OWNOFL	= 00000040	
COUNT	00000194	R	02	LKBSL_PARENT	= 00000048	
CREAD	000001EC	R	03	LKBSL_PID	= 0000000C	
CSID	0000023F	R	02	LKBSL_REMLKID	= 00000054	
CV_STRING	00000206	R	03	LKBSL_RSB	= 00000050	
CWRITE	000001EF	R	03	LKBSL_SQFL	= 00000038	
DIRENTRY	00000257	R	02	LKBSV_MSTCPY	= 00000004	
DIR_ENTRY	000002DB	R	03	LKBSW_FLAGS	= 00000028	
DISPLAY_LOCK	0000051D	RG	03	LKBSW_REFCNT	= 0000004C	
DMP1	000002CF	R	02	LKBSW_STATUS	= 0000002A	
DMP3	000002EF	R	02	LKB_BFR	00000008	R 02
DMP5	0000030F	R	02	LKB_RSB_BFR	0000006C	R 02
DMP7	0000032F	R	02	LKID	0000026F	R 02
DONE	00000442	R	03	LKID_BFR	000001A2	R 02
EMODE	000002F7	R	03	LKID_OFF	= 00000000	
EX	000001F8	R	03	LKMODE_TBL	000001D1	R 03
EXIT	00000F2A	R	03	LOCAL	000002BA	R 03
FAO_GROUP_DSC	000001AE	R	02	LOCKID	00000090	RG 02
FAO_RMINFO_DSC	00000094	R	02	LOCKMODE_TBL	000001C5	R 03
FLAGS1	00000277	R	02	LOCKSTR1	0000005A	R 03
FLAGS2	00000293	R	02	LOCKSTR2	0000006F	R 03
FLAGS3	000002AF	R	02	LOCKSTR3	00000093	R 03
FLAGS_TBL	0000022A	R	03	LOCK_COND_HAND	0000009E	RG 03
FORMAT_LOCK	00000556	RG	03	LOCK_COUNT	00000000	R 02
FORMAT_RSB	00000BCC	RG	03	LOCK_HEAD	00000000	R 03
GETLCK	000003C7	R	03	LOCK_STR_TBL	0000002E	R 03
GET_LKB	00000443	RG	03	MASTER	000002C8	R 03
GGMOD	000001EF	R	02	MODE1	0000028F	R 02
GRANT	= FFFFFFFF		MODE2	000002AB	R	02
GRANTSTR	00000141	R	03	MSG\$_LOCKIDZER	*****	X 03
GRMD_BFR	0000019C	R	02	MSG\$_NOLKB	*****	X 03
GRMD_OFF	= 00000005		MSG\$_NOLOCKS	*****	X 03	
GROUP_BFR	0000006C	R	02	MSG\$_NOPRLOCK	*****	X 03
GROUP_BUF	000001BA	R	02	MSG\$_NORESOURC	*****	X 03
GROUP_BUF_DSC	000001C0	R	02	MSG\$_OUTOFRANG	*****	X 03
GROUP_DSC	000001D8	R	02	MSG\$_SUCCESS	*****	X 03
GROUP_NUM	000001E8	R	02	NEW_PAGE	*****	X 03
			NOLCK	00000430	R	03

LOCK
Symbol table

LOCK AND RESOURCE FORMATTING ROUTINES

16-SEP-1984 01:31:38 VAX/VMS Macro V04-00
5-SEP-1984 03:32:46 [LSDA.SRC]LOCK.MAR;1

Page 41
(22)

NOT_VALID 000002D1 R 03
NSPACE 0000031F R 02
NULL 000001E9 R 03
NULL_CSTRING 00000222 R 03
PAGE_SIZE ***** X 03
PARID 00000283 R 02
PARID_BFR 00000004 R 02
PART2 0000067E R 03
PCBSL_LOCKQFL = 00000104
PID 00000273 R 02
PREAD 000001F2 R 03
PRINT ***** X 03
PRINT_LINE 00000F2B RG 03
PRINT_LOCK 00000856 RG 03
PRINT_RSB 00000D34 RG 03
PROCESS 000002C0 R 03
PROCESS_QUEUE 00000E55 RG 03
PRSB 000001F3 R 02
PWRITE 000001F5 R 03
QUEUE_COUNT 00000119 R 02
QUEUE_TYPE 00000198 R 02
QUE_STR_TBL 0000004E R 03
RACMOD 00000243 R 02
REQMEM ***** X 03
RES1 000002C7 R 02
RES2 000002C3 R 02
RES3 000002E7 R 02
RES4 000002E3 R 02
RES5 00000307 R 02
RES6 00000303 R 02
RES7 00000327 R 02
RES8 00000323 R 02
RESN1 0000021F R 02
RESN2 0000021B R 02
RESN3 00000233 R 02
RESN4 0000022F R 02
RESN5 0000024B R 02
RESN6 00000247 R 02
RESN7 00000263 R 02
RESN8 0000025F R 02
RESNAM_BFR 00000070 R 02
RESNLEN 0000022B R 02
RES_COUNT 00000115 R 02
RES_HEAD 00000015 R 03
RLEN 000002DF R 02
RMINFO 00000337 R 02
RMINFO_BFR 000000C5 R 02
RMINFO_CNT 000000C4 R 02
RMINFO_DSC 000000BC R 02
RMINFO_LEN 000000B8 R 02
RMOD_BFR 0000006E R 02
RNSPACE 0000025B R 02
RQMD_BFR 0000019F R 02
RQMD_OFF = 00000004
RSB = 000001EB R 02
RSB\$B_CGMODE = 0000000D
RSB\$B_GGMODE = 0000000C

RSB\$B_RMOD = 0000004E
RSB\$B_RSNLEN = 0000004F
RSB\$K_LENGTH = 00000050
RSB\$K_MAXLEN = 0000001F
RSB\$L_CSID = 00000038
RSB\$L_CVTQFL = 00000018
RSB\$L_GRQFL = 00000010
RSB\$L_PARENT = 00000048
RSB\$L_VALSEQNUM = 0000003C
RSB\$L_WTQFL = 00000020
RSB\$Q_VALBLK = 00000028
RSB\$T_RESNAM = 00000050
RSB\$V_DIRENTRY = 00000000
RSB\$V_VALINVLD = 00000001
RSB\$W_BLKASTCNT = 00000042
RSB\$W_GROUP = 0000004C
RSB\$W_REFCNT = 00000040
RSB\$W_STATUS = 0000000E
RSBCSID_BFR 00000068 R 02
RSB_BFR 0000011D R 02
RSNLEN_BFR 0000006F R 02
SAVE_LOCK_DATA 000004B7 RG 03
SEQNUM 00000213 R 02
SET_HEADING ***** X 03
SHOW_ALL_LOCKS 00000328 RG 03
SHOW_ALL_RES 00000A32 RG 03
SHOW_ONE_LOCK 0000036E RG 03
SHOW_ONE_RES 00000AF3 RG 03
SHOW_PROC_LOCK 000003AE RG 03
SHOW_QUEUES 00000B92 RG 03
SHOW_RSB 00000B62 RG 03
SKIP_LINES ***** X 03
SMODE 00000302 R 03
SPACE 000002CF R 03
SRBCT 000001FB R 02
SS\$RESIGNAL ***** X 03
SS\$UNWIND ***** X 03
STATE1 00000287 R 02
STATE2 000002A3 R 02
STATUS_TBL 00000272 R 03
STS1 000002D3 R 02
STS2 000002F3 R 02
STS3 00000313 R 02
SUBLKS 0000029F R 02
SYSS\$FAO ***** X 03
SYSSUNWIND ***** X 03
SYS_DSC 000001C8 R 02
TYPE 00000333 R 02
UMODE 0000030E R 03
VAL1 0000020F R 02
VAL2 0000020B R 02
VAL3 00000207 R 02
VAL4 00000203 R 02
VALID 00000217 R 02
WAIT = 00000001
WAITSTR 0000019B R 03
WT_STRING 00000214 R 03

MA
V04

4D

5F

46

4C

72

6D

20

59

4E

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SDADATA	0000033B (827.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
LOCK	00000F92 (3986.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
LITERALS	0000034C (844.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.24
Command processing	107	00:00:00.45	00:00:03.92
Pass 1	310	00:00:06.76	00:00:26.79
Symbol table sort	0	00:00:00.69	00:00:01.98
Pass 2	264	00:00:02.72	00:00:10.19
Symbol table output	26	00:00:00.14	00:00:00.30
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	741	00:00:10.83	00:00:44.44

The working set limit was 1650 pages.

66653 bytes (131 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 660 non-local and 73 local symbols.

1491 source lines were read in Pass 1, producing 79 object records in Pass 2.

26 pages of virtual memory were used to define 24 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1	9
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	21

660 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:LOCK/OBJ=OBJ\$:LOCK MSRC\$:LOCK/UPDATE=(ENH\$:LOCK)+EXECML\$/LIB+LIB\$:SDALIB/LIB

0352

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY